FACULTY
SCIENCE COMPUTER AND INFORMATION TECHNOLOGY

DEPARTMENT
ARTIFICIAL INTELLIGENCE

DESIGNING AN INTELLIGENT SYSTEM FOR GENETIC BIOLOGY LEARNING

S.GANGATHARAN A/L SEENIVASAGAM
WEK98274

THESIS II
(WXES3182)

SUPERVISED BY : ASSOCIATE PROFESSOR DR. SAPIAN BABA

MODERATED BY : PN. ROHANA MAHMUD

DATE OF SUBMISSION :12TH FEBRUARY 2001

ABSTRACT

With the changing education scenario, an issue which has come strongly on how to create intelligent tutoring system to educate learners in a more intelligent way. So a need has emerged for the development of an intelligent tutoring system.

This part of thesis describes the design of a collection of intelligent systems under the name GENETIC BIOLOGY for basic genetic biology tutoring. GENETIC BIOLOGY consist of three intelligent systems which are designed for cell, DNA and gene learning.

GENETIC BIOLOGY systems employ natural language processing and various techniques normally used in the construction of intelligent systems. Production rules and frames are widely used in representing knowledge of genetic biology.

ACKNOWLEDGEMENTS

Firstly, I must thank God Almighty for his mercy and grace throughout my studies because without him wouldn't have reach this far.

I also would like to acknowledge all those who helped and influenced me in this thesis, especially my supervisor, Associate Professor Dr. Sapian Baba and my moderator Pn. Rohana Mahmud. My appreciation is extended to Dr.Amir for is sharing of expertise in Genetics.

My gratitude also goes to my fellow course mates and friend for sharing their knowledge and idea with me in the field of intelligent system and natural language processing, especially to lecturers who runs the AI and Natural Language Processing classes .

I would like to extend my gratitude also to the Biology students, Jayaraj, Barath, Sabitha and others for giving ideas for the form of frequently asked questions in genetic biology. I also would like to thank my friend Shankar for always giving me a helping hand.

Finally, I also thank my beloved family members for their support in my academic life.

TABLE OF CONTENTS **Page**

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

This thesis focuses on the development of intelligent tutoring system for Genetic Biology using a combination of natural language processing and various existing techniques normally used in the development of such systems. Genetic biology is a complex and very technical area and it can be adequately presented with a computer-based system

Broadly, the primary goal of artificial intelligence is to make machines smarter by attempting to understand what intelligence is and to make machines more useful [Winston & Prendergast, 1984]. Intelligent system and expert system is a result of a 20-year quest to define the appropriate nature of such machine.

Intelligent system has several capabilities and features that distinguish it from conventional computer programs. Perhaps this can be viewed from their respective goals in which the basic one is to capture and distribute the expertise of human expert [Prerau, 1990]. In contrast, the goal of a conventional program is merely to implement a set of algorithms.

Recent developments have shown that the capabilities of intelligent systems could be improved when multiple knowledge engineering techniques under hybrid environment are incorporated [Silverman, 1987]. These techniques include multiple knowledge representation methods and the combination of qualitative and quantitative analyses. These are some of the major focuses in the development of intelligent systems described in this thesis.

Other conventional characteristics of intelligent system remain as are less influential in the development of the system. The following characteristics as described in Giarratano & Riley and Firebaugh are used as the general guidelines in the development of the system [Giarratano & Riley, 1989; Firebaugh, 1989].

1. *High performance.* The system must be able to respond at a level of competency equal to or better than expert in selected field. Simply, the quality of expertise from the system must be high.
2. *Highly domain specific.* The system is highly domain specific, that is, it knows a great deal about a narrow range of knowledge rather than knowing a little about everything.
3. *Adequate response time.* The system must be able to perform in a reasonable time, comparable to or even better than the time required by a human expert to arrive at a solution.
4. *Goal reliability.* The system must be reliable and not proned to crashes.
5. *Error handling.* The system should be able to handle and report error, since many applications involve uncertain information. In education, uncertainty may mean something which has not been discovered.

1.1 SCOPE AND OBJECTIVES OF RESEARCH

The broad objective of this thesis is to describe a collection of knowledge-based intelligent systems constructed for genetic biology analysis. This collection of knowledge known as Genetic Biology Intelligent System will assist learners to learn in knowing the inner construction of a cell, the construction of DNA, synthesis of protein and genetic engineering methods.

The main objective of this research are listed as follows:

1. *To design a collection of knowledge-based systems for a narrow sub-domain of a broad major domain area.*

   As mentioned earlier, Genetic biology is a wide area that requires different levels of presentation.

2. *To design and adapt representation of genetic knowledge which has both quantitative and qualitative characteristics.*

   The nature of genetic knowledge is both qualitative and quantitative. It is important that this knowledge is fully acceptable by the system for presentation. Thus it is in this aspect that a knowledge representation structure is designed to accommodate this type of knowledge.

3. *To study the accuracy of diagnosis of the presentation concerned.*

   As mentioned earlier, genetic information consist of technical diagnosis procedures. The system concerned must be able to handle such procedures.

To achieve the above aims, three separate modules are created. These systems are as follows:

1. CELLSYS - an intelligent system developed for presentation of cell components and functions.
2. DNASYS - an intelligent system that represent DNA components and steps for DNA construction.
3. GENSYS - an intelligent system for representing the gene expression and regulation. Show the flow of how a gene is copied and translated into protein.

Chapter 2

INTELLIGENT SYSTEM BACKGROUND AND LITERATURE REVIEW

The efficiency and performance of an intelligent system depend on the quantity and the quality of the system's knowledge. This chapter gives a general review of various methods for intelligent systems.

2.1 OVERVIEW OF INTELLIGENT SYSTEMS

Intelligent system is a branch of artificial intelligence [Bonnet, 1985; Charniak & McDermont, 1985; Harmon & King, 1985; Rauch-Hindin, 1988; and Waterman, 1986]. Artificial Intelligence (AI) is the man-made capacity to acquire and apply knowledge [Arnold & Bowie, 1986]. AI can be subdivided into many relatively independent research areas. One of these areas which is the focus of this thesis is intelligent tutoring system.

2.1.1 Human Experts and Intelligent System

Before an attempt is made to define an intelligent system, the term intelligent is examined.

> "An intelligent system is a system which able represent some intelligent of a person. This intelligence can encompass a wide area such as
> - understanding natural language
> - solving problem as an expert does
> - diagnose given facts to arrive to some solution or advise
>
> and etcs."

Based on the above definition, it is clear that to a certain extent, the development of intelligent systems relies considerably on the study of the human intelligent. Concept is extracted from human and it is then represented in the system. The system then attempts to emulate the performance of these intelligent. This whole process of building an intelligent system is known as knowledge engineering and intelligent systems are also referred to as knowledge system [Harmon & King, 1985].

However, it must be noted that the attempts of intelligent systems in evaluating the performance of human experts are limited in scope. Like a human expert, an intelligent system is a specialist focusing on a narrow set of problems or a domain. Its knowledge can be both theoretical and practical which is perfected through experience in the domain.

An intelligent system is not a complete cognitive modeling program. It does not attempt to simulate human mental architecture in detail. It is a practical program that uses natural language processing strategies developed by human to solve specific classes of problems.

2.1.2 Definiton

Having discussed the relationships between human intelligent and intelligent systems, some definitions of an intelligent system are examined in this section.

[Buchanan and Shortliffe [1984] define an intelligent system as an AI program designed to

a) provide expert-level solutions to complex problems,
b) be understandable, and
c) be flexible enough to accommodate new knowledge easily.

Feigenbaum [1982] has a more appropriate definition. He defines an intelligent system as:

> ..... an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. Knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of expertise of the best practitioners of the field.

Even though the term " intelligent system" has been widely used, it must be noted that sometimes terms such as "knowledge systems" and "knowledge-based systems" are also used [Walters et al., 1988; Mockler, 1989]. Although people have tried to distinguish between these terms to indicate the degrees of complexity, there are no widely accepted definitions. In general, many agreed that an intelligent system is a form of intelligent knowledge-based system.

2.2 THE DEVELOPMENT OF INTELLIGENT SYSTEM TECHNOLOGY

It is difficult to identify the first intelligent system because so many AI researchers were working on the same ideas at the same time [Waldrop, 1987]. Numerous intelligent systems emerged almost simultaneously. However, it is possible to divide the systems into two groups namely, "early systems" which were generally developed prior to 1980 and "recent systems" which include most systems developed since 1980 [Harmon & King, 1985].

In 1956, Newell, Shaw and Simon developed a program called Logic Theorist (LT) [Newell et al., 1963]. This program was developed under a joint project of the RAND Corporation and the Carnegie Institute of Technology. It is one of the earliest programs that examines the use of heuristics in problem solving. In fact, the term "heuristics" was referred by them as "the complex processes ...that are effective in problem-solving". However, as soon as the program was completed, it was criticized as being inefficient [Waldrop, 1987] and researchers went on to develop theorem-proving programs that can perform better.

In the following year, the same group of experts developed a system called General Problem Solver (GPS). The aim of the system is to get machines to solve problems requiring intelligence and developing a theory of how human beings solve such problems. The final version of GPS was described in depth by Ernst and Newell [Ernt et al., 1969].

2.2.1 MYCIN

The real landmark intelligent system is MYCIN, a program developed at the Standard Medical School of Stanford University. Initially implemented as a Ph.D. project of Shortliffe, MYCIN's task was to advise physicians on the diagnosis of blood and meningitis infections, and on the selection of drugs for treatment [Shortliffe, 1976].

MYCIN is the first program to have expertise acknowledged by the experts themselves. In one official test, a team of evaluators rated MYCIN's prescriptions correct in 65

percent of the cases, whereas the accuracy of human specialists given the same cases ranged from 42.5 to 62.5 percent [Harmon & King, 1985].

The inference engine used in MYCIN is similar to the broad reasoning principles used in GPS because it contains no information about diseases or treatments. The difference between the two systems is that GPS emphasizes on the reasoning principles themselves and the special-purpose knowledge about domain is assumed to be secondary. In MYCIN, as in other intelligent systems, the emphasis is on the knowledge base and not the inference engine [Waldrop, 1987].

2.2.2 PROSPECTOR

One of the most notable intelligent systems created in the late 1970s is PROSPECTOR. Created at the Stanford Research Institute International (SRI), the system is designed to advise geologists on whether or not a given site might have pre-grade deposits [Duda et al., 1984]. This system has features quite similar to those of diagnosing a disease, except that it evaluates the geological characteristics of a mining site. In 1980, the system had successfully identified an ore-grade molybdenum deposit at Mount Tolman, in Washington State. This makes the system the first intelligent system to have a major commercial contribution [Waldrop, 1987].

2.2.3 CADUCEUS

Another intelligent system created in the early 1970s is CADUCEUS. The first version of this system is known as INTERNIST-I. This system is the product of d.E. Pople and J.D. Myers of the University of Pittsburgh [Winston & Prendergast, 1984]. The system covers some five hundred diseases in its knowledge base and this covers close to 25 percent of internal medicine as a whole. Knowledge in the system is represented as a network of findings and diseases and is assessed based on the constraints of the disease taxonomy and causal relationship [Pople, 1984].

2.3 INTELLIGENT SYSTEM IN EDUCATION AND RELATED AREAS

There have been many reviews of educational intelligent systems. Such reviews covers a broad area of education-related areas ranging from engineering, electronic, medicine, tutoring, athletes training assist system, geometry and even disability learning system. Selected systems related to these areas are summarized in the following subsections. Some of the features found in many of these systems are used in the construction of the systems discussed in this thesis.

2.3.1 ATEC

ATEC is an intelligent tutoring system that assists in the training of air traffic controllers, without subjecting them to the hazard of real conditions where a single mistake could prove disastrous. The system is part of the Intelligent Simulation Training System (ISTS) project. ISTS approaches the problem by combining a computer-based graphic simulation of radar scope intelligent tutoring module, and an intelligent system that represents the knowledge of an expert air traffic controller (ATC). A variation of the ATEC system could alternatively be used in connection with a real radar scope to assist a working controller in everyday decisions. The system was developed at the Department of Computer Science, Embry-Riddle Aeronautical University, Daytona Beach, FL.

2.3.2 COACH TRAINING ASSISTANT APPLICATION

An intelligent system was developed to assist a coach with training athletes. The system models selected aspects of a master coach thereby providing the novice coach with insights into the game that would normally take years to acquire. It includes access to a laser video disc that permits the portrayal of the skills and tactics, training progressions, and other information used by the master coach so that the technical complexity inherent in physical movement can be controlled, verbally described, and graphically presented to the user. The system was developed at Calgary University, Alberta, Canada.

2.3.3 DIGITS

DIGITS is an intelligent computer-aided instruction system for tutoring of an introductory course in digital electronics. DIGITS functions by guiding the student

towards the discovery of issues in the tutorial material. DIGITS was implemented in ADA at Case Western Reserve University in Cleveland, OH.

2.3.4 GUIDON

GUIDON is an intelligent tutoring system in the area of medicine. The system is an adaptation of MYCIN, which offers the student a tutor for medical diagnosis. GUIDON provides the student a medical case to solve and checks the student's understanding using the original diagnostic rules used in MYCIN. The student asks questions to gather important data and proposes hypotheses. The system then compares the student's behavior to the expert behavior as modeled within MYCIN's rules. It also intervenes when student requests help or when the student runs into trouble. The system is rule based and is implemented in LISP.

2.3.5 LEARNING DISABILITY APPLICATION

An intelligent system was developed to assist a child with a learning disability. When a child is experiencing learning difficulties the exact nature of the child's problems must be determined in order to plan a successful instructional program. This system guides the teacher through the various stages of diagnosing learning disabilities. Specifically, it works to assist in the assessment of reading problems. The system is implemented in LISP and was developed at the University of Saskatchewan, Canada.

2.3.6 MENTONIEZH

MENTONIEZH is an intelligent tutoring system in geometry that coaches and corrects a student during two stages of problem solving: figure drawing and proof building. The system uses a plan recognition method, which deduces, from the problem's space research and the student's inputs, the underlying intention, and can also detect intention shifting. It relies on the classical architecture of an ITS. The expert of the domain - a set of geometry rules together with an inference engine - can solve problem in different ways and distinguish the student's erroneous or unuseful inference. The pedagogical model is used to conduct the interaction between the student and the system, to give explanations

if the student applies an erroneous reasoning, and to provide him with clues when in deadlock. The system was developed at RISA/INRIA, Rennes, France.

2.4 ARCHITECTURE OF AN INTELLIGENT SYSTEM

Figure 2.1 provides an overview of the architecture of an intelligent system. The user interacts with intelligent system through a user interface that makes access more comfortable for the human and hides much of the system's complexity. The interface employs a variety of interface styles, including question and answer, menu driven, natural language, or graphics interfaces.

The knowledge base is the portion of an intelligent system that contains the domain knowledge. The knowledge is represented by facts(data) and rules (or other representations) that are used as the basis for decision making. This knowledge is normally obtained from one or several human experts.

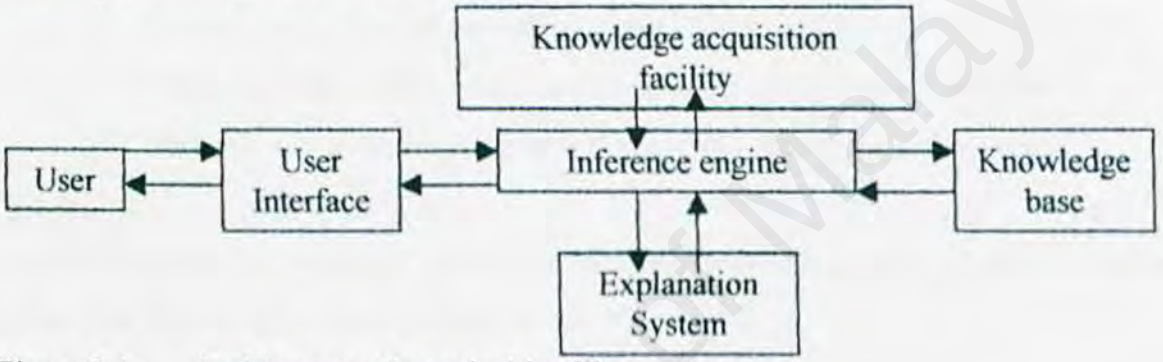

Figure 2.1 Architecture of a typical intelligent system

The knowledge acquisition facility provides a dialogue between the system and the user in which knowledge is acquired and stored in the database. It assists in the acquisition of new knowledge, helps in maintaining correct rule syntax, and checks on the updated knowledge base to ensure that consistency is maintained. An example of such a facility is the Teiresias knowledge base editor [Davis & Lenat, 1980].

The inference engine is the interpreter for the knowledge base. It performs the inference reasoning tasks for the intelligent system by using the knowledge stored in the knowledge base and information provided by the user.

2.5 THE ROLE OF KNOWLEDGE REPRESENTATION

Knowledge representation (KR) plays an important role in intelligent systems. However, its role is often difficult to define. In some cases, KR only manages a collection of data structures while in others, KR does almost everything: making decisions, proving theorems and so on. In this section, the general role of KR within an intelligent system is discussed.

Smith defines a knowledge representation hypothesis as [Smith, 1985]:

> "Any mechanically embodied intelligent process will be comprised of structural ingredients that
>
> (a) we as external observers naturally take to represent a prepositional account of the knowledge that the overall process exhibits, and
>
> (b) independent of such external semantically play a formal but causal and essential role in engineering the behavior that manifest that knowledge."

In this section, the methods defined above are presented. These methods include production rules, logic, frames, semantic nets and objects.

2.5.1 Production Rules

The production rules techniques in knowledge representation is one of the most commonly used techniques in AI. They are also referred to as if-then pairs, situation-action pairs or just simply productions.

This approach of representation knowledge was first introduced by E.L. Post [Post, 1943]. However, the early proposal has undergone much theoretical and applications-oriented change in the years. The modern application of the production rules is mainly the work of Newel and Simon [Newel & Simon, 1972]. They used the production systems for their models of human cognition.

In rule-based system, the expertise of the problem area is represented by the productions with the premises of the rules (the IF portion) corresponding to the action. An example of

such rules is given in figure 2.2. The rule contains the premise or the condition in the "if" clause and the conclusion in the "then" clause.

```
Rule 1: IF transformation is first
        AND translation is next
        THEN accept
```

Figure 2.2. An example of a IF-THEN rule

2.5.2 Logics

Formal mathematical logical systems were some of the first representation schemes to be used in the building of intelligent systems [Schutzer, 1987]. First-order logic and predicate calculus are currently the most commonly used mathematical logic systems. Using this formalism, a new statement can be concluded true by proving that it follows from statements already known to be true.

The most fundamental concept in logic is that of truth. A properly formed statement (or proposition) has only one of two possible values: true or false. Propositional logics adds to this concept of truth of various connectives, or logical operators. These are listed in the Table 2.1.

Table 2.1 List of logical operators

AND	^ or &
OR	v
NOT	~
IMPLIES	---> or >
EQUIVALENT	= or <--->

The above connectives are defined by the truth table shown below [Barr et al., 1981(a)]. In the table, t stands for true, f stands for false, and p and q represents two propositions.

Table 2.2 Truth table to define connectives

p	q	p^q	pvq	p → q	~p	p<-->q
t	t	T	t	t	f	t
t	f	F	t	f	f	f
f	t	F	t	t	t	f
f	f	F	f	t	t	t

In propositional calculus, the first rules of inference are encountered. An inference rule allows the deduction of new sentences from previously given sentences. One of the main forms of inference is modus ponens which states that:

from (if p → q) and p, then infer q

Prolog is the first practical and widely available logic programming language. Logical expressions can also be implemented in other programming languages such as LISP and Pascal.

2.5.3 Frames

Frames are a kind of template for holding related clusters of data, facts, rules, hypotheses or any other knowledge in a single conceptual unit. These frames require conceptual reorganization of situation data around key controlling concepts.

An example of a frame is given in Figure 2.3.

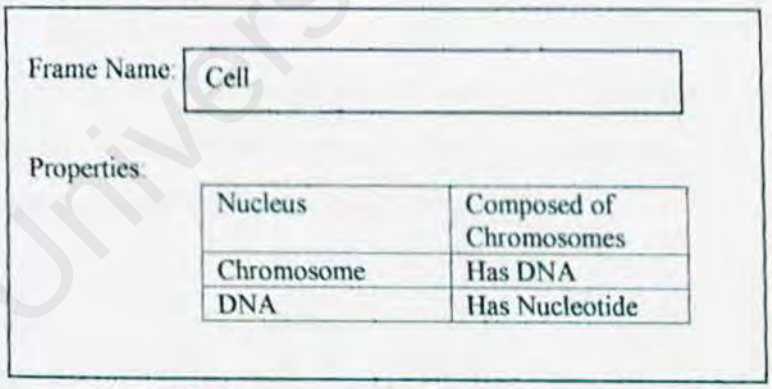

Figure 2.3 An example of a frame.

The idea of frame was first introduced by Minsky which he defines as "a data structure that includes declarative and procedural information in a predefined internal relation"

[Minsky, 1975]. According to Minsky, a frame may be viewed as a static data structure used to represent well-understood stereotyped situations.

Frames use slots to hold information about an item or object. This information usually covers its attributes and values as well as procedures and pointers for getting further information from other frames in the expert system. Frames may also contain rules and questions involving information in frames. Another unique feature of this approach is the use of default value, which is a value assumed when no other value is available.

Frame-based systems enable dealing with more complex situation than can be dealt with in systems that contain just rules and questions. The psychological understanding of this approach is that humans think in clusters, especially in decision situations. Thus the frame approach is closer to mimicking the ways human beings reason and remember.

2.5.4 Semantic Networks

Semantic network is a knowledge representation based on a network structure [Quillian, 1968; Raphael, 1968]. This method was developed originally as a psychological model of human memory but has now been adopted as major knowledge representation method in artificial intelligence.

A semantic net consists of points called nodes connected by links called arcs which describe the relations between the nodes. The nodes in a semantic net represent objects, concepts or events. Arcs used to represent hierarchies include "is-a" and "has-a" parts. Figure 2.4 shows an example of a semantic network.

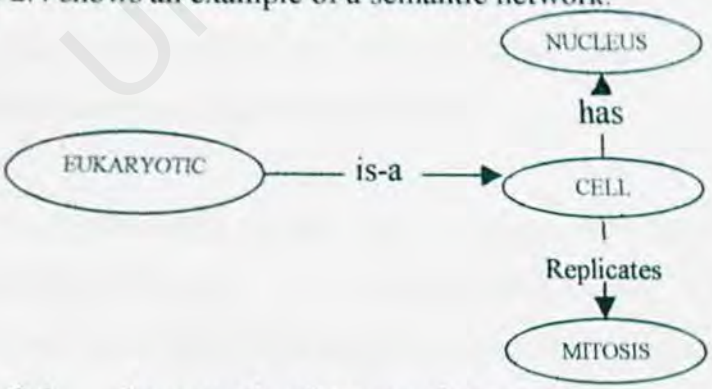

Figure 2.4 An example of a semantic network

2.5.5 Object-Oriented

Objected-oriented knowledge representation is a relatively new method for designing and implementing intelligent systems. It centers around several major concepts such as abstract data types and classes, type hierarchies (subclasses), inheritance and polymorphism [Pinson et al., 1988].

The basis of this method of representation is the ability to define computational objects of arbitrarily complex internal structure, which may be thought of subsequently as a single entity. These computational objects are not passive data structures (such as frames); they encapsulate both state and behavior. Their behaviors are implemented as attached procedures, called methods. These are procedures related to an object. They are distinguished from standard procedures in that they are stored in (or inherited by) an object's data structure and are invoked when the object receives a message from another object.

Objects are active in the sense that the methods are bound to the object itself, rather than existing as separate procedures for the manipulation of a data structure. Although frames allow the creation of complex objects and the integration of procedural and declarative representations, they are passive data structures that must be acted on by external procedures.

Another major feature of object-oriented programming languages is the support of classes and inheritance. In a pure object-oriented system, everything is an object and all objects belong to classes. This allows inheritance of slot names, values and methods. In addition, each class object defines instance variables, which must be instantiated when an individual member of that class is created.

A pure object-oriented system is one in which everything is an object. An example of such a system is SMALLTALK [Goldberg et al., 1983]. A hybrid system is one in which objects coexist within a conventional programming language [Tello, 1989]. Examples of

this include C++ [Wiener et al., 1988] and CLOS [Keene, 1989]. These are extensions from C and LISP respectively.

2.6 INFERENCING STRATEGIES

There are many strategies in inferencing within an intelligent system. For the sake of discussion, two most common strategies used in constructing intelligent systems are discussed. Discussions on other strategies can be easily obtained from other sources [Winston, 1984; Martin & Oxman, 1988].

2.6.1 Backward Chaining

Backward chaining works from the goal state to the initial state. It is a goal directed search and attempts to find the data to prove or disprove a hypothesis. Thus backward chaining systems are also known as goal-directed systems. This is possible if the goals are known and if they are reasonably small in number.

To illustrate how this reasoning mechanism is used, consider the following set of rules:

Rule

1. IF DNA polymerase attached to parental DNA
   AND the stop codon is encounted
   THEN polymerase reach end
2. IF polymerase reach end
   THEN polymerase detaches
3. IF polymerase detaches
   THEN daughter DNA is produced

Suppose that the objective is to establish the fact that "daughter DNA is produced" given the facts that the enzyme helicase separate DNA double helix and DNA polymerase attached to parental DNA. The backward-chaining method works backward from the conclusion.

Is the fact known?	>	No
Can it be obtained from a rule?	>	Yes, by rule 3
Which fact(s) needs to be known?	>	" polymerase detaches "
Is this fact known?	>	Yes, by rule 2
Which fact(s) needs to be known?	>	"polymerase reach end"

Is the fact known?	>	No
Can it be obtained from a rule?	>	Yes, by rule 1
Which fact(s) needs to be known?	>	"DNA polymerase attached to parental DNA" and "the stop codon is encountered"
Are these facts known	>	Yes

There, it is true that: " polymerase reach end"

There, it is true that: " polymerase detaches"

There, it is true that: " daughter DNA is produced"

The above clearly shows that given a fact to prove (the goal), the mechanism tried to establish all the facts needed to reach that goal. The reasoning method is called backward chaining. It is applied when a goal or hypothesis is chosen as the starting point for problem solving.

2.6.2 Forward Chaining

Forward chaining works forward from current state to the goal state. The goal or solution needs to be constructed or assembled. The number of outcomes is large. It is essentially a data driven strategy and may lead to unfocused search or reasoning. Forward chaining systems are sometimes called data-driven or fact-directed reasoning [Bonnet et al., 1988].

The forward-chaining reasoning mechanism goes forward from antecedents to the conclusion they generate. Suppose that the fact " daughter DNA is produced" is to be proven, given the facts that " polymerase attached to parental DNA" And "the stop codon is encounted ".

Is the fact known?	>	No
Which facts do we know?	>	" polymerase attached to parental DNA" and "the stop codon is encountered"
What facts follow from it?	>	" polymerase reach end" by rule 1
Is the what we want to prove?	>	No
What facts follow from this?	>	" polymerase detaches " by rule 2

Is the what we want to prove? > No

What facts follow from this? > "daughter DNA is produced" by rule 3

Is the what we want to prove? > Yes

From the facts that " polymerase attached to parental DNA" and "the stop codon is encountered", the fact "polymerase reach end" is esteblished. From the fact that " polymerase reach end", the fact " polymerase detaches " is established. Finally, from the fact "" polymerase detaches ", the fact "daughter DNA is produced" is proven.

The forward chaining reasoning mechanism is suitable in solving problem in which data is to be used as the starting point for problem solving. A combination of forward chaining and backward chaining can be used to get to problem solution quickly when dealing with large search spaces.

2.7 DEALING WITH UNCERTAINTIES

Uncertainties arises constantly in practical situations. Real life problem-solving demands an acceptance of uncertainty, in order to minimize the difficulties it poses. Rolston classified uncertainty that are common in expert domains into the following groups [Rolston, 1988]:

1. Uncertain knowledge.

   Many a time the expert will have only heuristic knowledge regarding some aspect of the domain. For example, given only a set of evidence, the expert has to arrive at a conclusion.

2. Uncertain data.

   Even when an expert is certain of the domain knowledge, there may still be uncertainty in the data that describes the external environment. For example, in the attempt to infer a specific cause from an observed effect, the expert may have to rely on questionable test results.

3. Incomplete Information.

   Frequently the expert has to make decisions based on incomplete information. For example, he must make such decisions in the course of processing incrementally.

4. Randomness

   Some domains are inherently random; even though the available knowledge and information is complete and the knowledge is certain, the domain still has stochastic properties.

Intelligent system must be able to reason under uncertainty in order to solve a large number of real-world problems. Various techniques have been developed which allow the use of fragmentary and uncertain information to reach an estimate of the truth. The following are some of the methods developed to manage uncertainty that results from heuristic rules:

(a) Bayesian Probability Theory

(b) Stanford Certainty Approach

(c) Fuzzy Set Theory

(d) Denmpster/Shafer Theory of Evidential Reasoning

(e) Nonmonotonic Reasoning

It must be noted that these methods will not completely duplicate human capabilities but each has proved to be useful in the development of intelligent systems.

2.8 SUMMARY

This chapter describes the basic concepts and techniques of building intelligent systems. The description starts with a brief history on the development of intelligent/expert technology. This is followed by a survey of existing intelligent systems developed for education and education -related applications. An overview of intelligent system architecture is presented and five methods for representing knowledge are described. Methods for inferencing strategies and methods dealing with uncertainties are also described.

Among others, the intelligent systems developed under the GENETIC BIOLOGY project employees some ideas discussed in this chapter. Handling uncertainty and heuristic reasoning are not employed to the knowledge domain of genetic. These are explained in the following chapters.

Chapter 3

FRAMEWORK OF GENETIC BIOLOGY

The GENETIC BIOLOGY project deals with the applications of intelligent system technology in genetic biology with particular reference to cell contents and functions, DNA components and construction procedures, protein synthesis steps and genetic engineering methods. Specifically the system is constructed to educate the genetic related topics of a cell, DNA and genes.

The system guides the student in learning genetics. The guidance are built information and knowledge obtained from experts and books in related topics. The system infers the students' queries. It uses natural language processing techniques to process students query. It then compare the student model of query with an "its own" model give solution. This task is then followed by remedial instruction to correct any unknown words or misspelling .

The intelligent system is designed to allow the user, whose knowledge is likely to comprise "chunks" of information, to integrate and expand such knowledge within the larger framework of useful knowledge and methods used by an expert. Thus the system reflects the expertise in explanation under the genetic environment.

In this chapter, the framework of genetic analysis is presented. This framework provides the fundamental foundation for the construction of the intelligent system of our GENETIC BIOLOGY project.

3.1 THE KNOWLEDGE DOMAIN OF GENETIC BIOLOGY

Studies into the nature of genetic biology have highlighted some important differences between an experts knowledge and that of novice. Experts are schema driven. They possess knowledge which specifies both the categories to which a problem belongs and the appropriate solution moves. The type of knowledge is at a deeper, more principled, level than that of the novice. It is these deeper level of knowledge which experts often have greatest difficulty in articulating.

Acquiring knowledge from an expert and representing it within an intelligent system is a multi-faceted technical, cognitive and psychological process. On the one hand, the knowledge engineer must acquire sufficient knowledge of the domain to be able to communicate intelligently with the domain expert by, for example, adopting a creative listening approach: searching for the essential underlying concepts and reflecting these in words or diagrams back to the domain expert for critique or verification. The domain expert, on the other, needs to acquire an understanding of the structure imposed by the computational requirements on the way that knowledge must be framed.

3.1.1 Enhancement Of The Expert's Knowledge

The knowledge enhancement methodology used in this project is based on the following ideas:

1. to help the domain expert fill any knowledge gap.
2. to ensure that students needs for the knowledge of genetic biology was reflected in the intelligent system, and
3. to ensure that the intelligent system would fit within the existing user abilities.

3.1.2 The Knowledge

The framework that can be applied in the presentation of genetic biology is shown in

Figure 3.1.

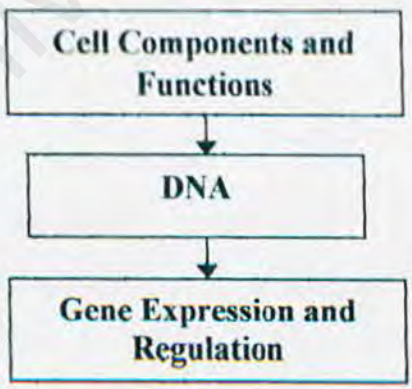

Figure 3.1 Components of Genetics Presentation

3.2 CELL COMPONENT AND FUNCION

Every living organism is made up of one or more cells. The smallest living organisms are single cells and cells are the functional units of multicellular organisms. All cell arise from preexisting cells. There two basic type of cell.

1. Prokaryotic for bacteria and archaeans.
2. Eukaryotic for plants and animals.

Cell are divided into three structures:

<table>
<tr><td colspan="4">1 Cell surface</td></tr>
<tr><td colspan="4">Composed of :</td></tr>
<tr><td></td><td colspan="3">1. Cell wall</td></tr>
<tr><td></td><td colspan="3">Function: Which function is to protect and support the cell.</td></tr>
<tr><td></td><td>Prokaryotic</td><td>Plants</td><td>Animals</td></tr>
<tr><td></td><td>Present</td><td>Present</td><td>Absent</td></tr>
<tr><td></td><td colspan="3">2. Plasma Membrane</td></tr>
<tr><td></td><td colspan="3">Function: Isolates cell contents from enviroment;
Regulates movement of materials into and out of cell;
Communicates with other cells.</td></tr>
<tr><td></td><td>Prokaryotic</td><td>Plants</td><td>Animals</td></tr>
<tr><td></td><td>Present</td><td>Present</td><td>Present</td></tr>
<tr><td colspan="4">2. Organization of genetic material</td></tr>
<tr><td colspan="4">Composed of :</td></tr>
<tr><td></td><td colspan="3">1. Genetic material</td></tr>
<tr><td></td><td colspan="3">Function: Encodes information needed to construct cell and control cellular activity</td></tr>
<tr><td></td><td>Prokaryotic</td><td>Plants</td><td>Animals</td></tr>
<tr><td></td><td>DNA</td><td>DNA</td><td>DNA</td></tr>
<tr><td></td><td colspan="3">2. Chromosomes</td></tr>
<tr><td></td><td colspan="3">Function: Contain and control use of DNA</td></tr>
<tr><td></td><td>Prokaryotic</td><td>Plants</td><td>Animals</td></tr>
<tr><td></td><td>Single, circular,
no protein</td><td>Many, linear,
with protein</td><td>Many,linear,
with protein</td></tr>
</table>

	3. Nucleus		
	Function: Membrane-bound container for chromosomes		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
	4. Nuclear envelope		
	Function: Encloses nucleus		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
	5. Nucleolus		
	Function: Synthesizes ribosomes		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
3. Cytoplasma structure			
Composed of :			
	1. Mitochondria		
	Function: Produce energy by aerobic metabolisn		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
	2. Chloroplasts		
	Function: Perform photosynthesis		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
	3. Ribosomes		
	Function: Provide site of protein synthesis		
	Prokaryotic	Plants	Animals
	Present	Present	Present
	4. Endoplasmic reticulum		
	Function: Synthesizes membrane components and lipids		
	Prokaryotic	Plants	Animals
	Absent	Present	Present

	5. Golgi complex		
	Function: Modifies and packages proteins and lipids; Synthesizes carbohydrates		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
	6. Lysosomes		
	Function: Contains intracellular digestive enzymes		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
	7. Plastids		
	Function: Store food pigments		
	Prokaryotic	Plants	Animals
	Absent	Present	Absent
	8. Central vacuole		
	Function: Contains water and wastes; Provides turgor pressure to support cell		
	Prokaryotic	Plants	Animals
	Absent	Present	Absent
	9. Other vesicles and vacuoles		
	Function: Contains food obtained through phagocytosis; Contain secretory products		
	Prokaryotic	Plants	Animals
	Absent	Present(some)	Present
	10. Cytoskeleton		
	Function: Gives shape and support to cell; Positions and moves cell parts		
	Prokaryotic	Plants	Animals
	Absent	Present	Present
	11. Centrioles		
	Function: Synthesize microtubules of cilia and flagella; May produce spindle in animal cells		
	Prokaryotic	Plants	Animals
	Absent	Absent(in most)	Present

	12. Cilia and flagella		
	Function: Move cell through fluid or move fluid past cell surface		
	Prokaryotic	Plants	Animals
	Present	Absent(in most)	Present

3.3 DNA

3.3.1 Composition Of Choromosome

- DNA is the composition of chromosome in strands.
- A gene is a functional segment of DNA located at a particular place on a chromosome.
- DNA or deoxyribonucleic is a nuclei acid consists of four very similar subunit of nucleotides.
- Each nucleotide contains deoxyribose.
- Each nucleotide of DNA consists of three parts:-
  i. A phosphate group
  ii. Deoxyribose
  iii. A nitrogen-containing base that has a single-ringed or double-ringed structure:
- Four types of nucleotides have the same phospate and sugar but different base.
- The pyrimidine bases thymine (abbreviated T) and cytosine (C) have single ring:-
- The purine bases adenine(A) and guanine(G) have double rings:-
- Sequence of bases in DNA encode vast amount of information.
- Four types of bases can be arranged in any linear order along a strand of DNA.
- Each sequence of bases represents unique set of genetic instructions.

3.3.2. Structure Of Dna

DNA of Chromosome Is a Double Helix

- DNA molecule consists of two strands, each composed of a series of nucleotides.
- In each single strand of DNA, the phosphate group of one nucleotide bonds to the sugar of another nucleotide.
- This bonding pattern forms a long strand consisting of a "backbone" of alternating sugars and phosphates, with the bases protruding from backbone.
- The DNA strand has a "free" sugar on one end and a "free" phosphate on the opposite end.
- The two strand of nucleotides are twisted about each other into a double helix, much like a ladder twisted lengthwise into a corkscrew shape.

- The sugar -phosphate back-bones of the two DNA strands are on the outside of the double helix, like the uprights of the ladder.
- Notice the sugar-phosphate uprights run in opposite directions, so that one upright has its free sugar on one end of DNA molecule while the other has its free sugar on the opposite end.
- The bases are packed into the middle, paired up to form the rung of the ladder.

The "arrowheads"(with the oxygen at the point of the arrow) represent the sugar, the circles represent the phosphate groups. Hidrogen bonds between complementary base pairs are indicated by the dashed line.

- Specific patterns of hydrogen bonding allow complementary bases to pair together in the center of the helix.
- Three hydrogen bonds hold guanine to cytosine.
- Two hydrogen bonds hold adenine to thymine.
- In nucleic acids, bases that pair together, held by hydrogen bonds, are called complementary base pairs.
- In DNA, adenine is complementary to thymine and guanine is complementary to cytosine. This is called "base-pairing rule".

3.3.3. Dna Replication And Its Genetic Constancy

- When a chromosome is duplicated, its double-helical strand of DNA is copied in a process called "replication" to two produce two identical DNA double helices.
- A chromosome could be replicated by separating the two DNA strands and synthesizing new strands from nucleotides with bases complementary to the parental strands.
- Each new chromosome would then consist of the one parental strand and one complementary daughter strand.

DNA Replication Occurs in Three Basic Steps

The DNA replication in all cells occurs in three fundamental steps, each catalyzed by an enzyme the :The two original, or parental, DNA strands of double helix unwind and separate.

i. Each parental strand is used as template for the formation of new daughter strand is formed by connecting nucleotides in an order determined by the nucleotide sequence of the parental strand according to the base-pairing rule: Adenine pairs with thymine and cytosine pairs with guanine.

ii. Finally, one parental DNA strand and its newly synthesized daughter strand wind together into one double helix, while the other parental strand and its daughter strand wind together into a second double helix. In forming a new double helix, the process of DNA replication converses one parental DNA strand and produces one newly synthesized strand. The process is called semiconservative replication.

Unwinding Segments of The Parental DNA Strands

- The first steps is to separate the parental DNA strands.
- An enzyme name DNA helicase(an enzyme that breaks apart the helix) works its way between the two strands, breaking the hydrogen bonds that hold them together.
- The DNA helicase then "walks"
- The result is that the two DNA strands separate, thus exposing their bases.

Complementary Daughter DNA Strands Are Synthesized

- The second step of DNA synthesis requires the enzyme DNA polymerase .
- This enzyme is responsible for joining nucleotide subunits to form the new strand of DNA.
- Two DNA polymerase molecules now bind to the unwound strands, one to each strands.
- During the replication, DNA polymerase performs a dual function.

- First, it recognizes bases exposed in parental DNA strand, and matches them up with free nucleotides that have complementary bases: Adenine forms hydrogen bonds with an exposed thymine from parental DNA strand and guanine with an exposed cytosine.
- Second, DNA polymerase bonds together the sugars and phosphates of the complementary nucleotides to form the backbone of the daughter strand.
- DNA polymerase can travel only in one direction on a DNA strand, from the "free sugar" end to the "free phosphate" end.

- Thus, the two DNA polymerase molecules, one on each parental strand, move in opposite direction. This process necessitates a third step of DNA synthesis.
- As the DNA helicase molecule continue to separate the parental DNA strands, one polymerase molecule follows behind it, synthesizing a long, continuous complementary daughter strand as it goes.
- The polymerase that settled on the second strand also moves in the "free sugar" to "free phosphate" direction; thus the enzyme travels away from the DNA helicase.
- Therefore, as the helicase continues to separate the parental strands, this polymerase cannot reach the newly separated parts of the second strand.
- Soon, however, a new DNA polymerase molecule attaches to the second strand close behind the helicase.
- Like the first polymerase, it moves away from the helicase.
- When the this second polymerase reaches the place where the first polymerase started, the two polymerases must be joined to make a continuous DNA strand.
- An enzyme called DNA ligase bonds the two strands together.

This process is repeated perhaps 10 million or so times for a human chromosome until the second strand has been replicated.

3.4 GENES EXPRESSION AND REGULATION

3.4.1 How Are Genes and Proteins Related

- A gene is a segment of DNA whose sequence of nucleotides specifies the sequences of amino acids in a particular protein.
- Each chromosome, which contains one long DNA molecule, includes many thousands of genes.
- Most genes store information that used to produce a protein.
- Some genes carry information for the synthesis of structural proteins, such as collagen in skin or keratin in hair.
- Few genes, the final product isn't protein but a nucleic acid called ribonucleic acid (RNA).

3.4.1.1 Genes Encode the Information for the Synthesis of a protein

As generalization, each gene encodes the information for a single protein; this generalization is called the **one-gene, one-protein hypothesis**.

- Different genes have different base sequences and different proteins have different amino acid sequences.
- Therefore, the sequence of bases in DNA encodes the sequence of amino acids in a protein.

3.4.1.2 How does a cell convert information stored in its DNA base sequences into protein?

DNA provides instruction for protein synthesis via RNA intermediaries.

- DNA located in cell's nucleus.
- Protein synthesis occurs on ribosomes in the cytoplasm.
- Therefore, DNA cannot directly guide protein synthesis.
- So, RNA act as an intermediary which carries information from DNA in the nucleus to ribosomes in the cytoplasm.
- RNA similar to DNA but differs structurally in three respects:
  1. RNA is normally single-stranded
  2. RNA has sugar ribose instead of deoxyribose in its backbone.

3. The base uracil in RNA replaces thymine in DNA.

- There three types of RNA in a cell
  1. messenger RNA (mRNA):carries the code from genes to ribosomes.
  2. Ribosomal RNA (rRNA):combines with protein to form ribosomes, on which protein synthesis occurs.
  3. Tranfer RNA (tRNA):carries amino acids to ribosome.

Table 3.1 A Comparison of DNA and RNA

	DNA	**RNA**
Number of Strands	2	1
Type of Sugar	deoxyribose	Ribose
Base Pairs	Adenine(A)-Thymine(T) Cytosine(C)-Guanine(G)	Adenine(A)-Uracil(U) Cytosine(C)-Guanine(G)

3.4.1.3 Genetic Information Flows in a Cell from DNA to RNA in Order to Direct Protein Synthesis

- Information from DNA is used to direct synthesis of proteins in two-step process .
  1. In transcription, the information contained in the DNA of a specific gene copied into messenger RNA (mRNA). The sequence of bases in messenger RNA carries information from nucleus to the ribosomes. This information specifies the sequences of amino acids in the protein to be manufactured.
  2. In translation, tranfer RNA (tRNA) and ribosomal RNA (rRNA) convert the information of the base sequence in messenger RNA into a specific amino acid sequence and thereby help synthesize the protein.
- The genetic code uses three bases to specify each amino acid
- There are four types of bases in DNA and four types of bases in RNA (Table 3-4-1).
- However, there are 20 different amino acids in proteins.
- Therefore, one base cannot code for just one amino acids because there are not enough types of bases.

- The genetic code must rely on short sequence of bases to encode each amino acids.
- Three bases specify a single amino acid.
- So, genetic code is hypothesized to be triplet code.
- These mRNA triplets are called codons that code for each amino acid (Table 3.4).
- One mRNA contains thousands of bases.
- So, to recognize where the code for given protein starts and stops.
  - The start codon is AUG.
  - The stop codon are UAG, UAA, and UGA.
- The start codon codes for amino acid that signals the beginning of the protein whereas the stop codon signals the ribosome to release the mRNA and the new protein.
- There 60 codons but only 20 amino acids for which to code. All 60 codons are used in the genetic code. Thus, a single amino acid may be specified by several codons.

Table 3.2 The Genetic Code(Codons of mRNA)

		Second Base									
		U		C		A		G			
First Base	U	UUU	Phenylalanine	UCU	Serine	UAU	Tyrosine	UGU	Cyteine	U	Third Base
		UUC	Phenylalanine	UUC	Serine	UAC	Tyrosine	UGC	Cyteine	C	
		UUA	Leucine	UCA	Serine	UAA	Stop	UGA	Stop	A	
		UUG	Leucine	UCG	Serine	UAG	Stop	UGG	Tryptophan	G	
	C	CUU	Leucine	CCU	Proline	CAU	Histidine	CGU	Arginine	U	
		CUC	Leucine	CCC	Proline	CAC	Histidine	CGC	Arginine	C	
		CUA	Leucine	CCA	Proline	CAA	Glutamine	CGA	Arginine	A	
		CUG	Leucine	CCG	Proline	CAG	Glutamine	CGG	Arginine	G	
	A	AUU	Isoleucine	ACU	Threonine	AAU	Asparagine	AGU	Serine	U	
		AUC	Isoleucine	ACC	Threonine	AAC	Asparagine	AGC	Serine	C	
		AUA	Isoleucine	ACA	Threonine	AAA	Lysine	AGA	Arginine	A	
		AUG	Start(Methionine)	ACG	Threonine	AAG	Lysine	AGG	Arginine	G	
	G	GUU	Valine	GCU	Alanine	GAU	Aspartic Acid	GGU	Glycine	U	
		GUC	Valine	GCC	Alanine	GAC	Aspartic Acid	GGC	Glycine	C	
		GUA	Valine	GCA	Alanine	GAA	Glutamic Acid	GGA	Glycine	A	
		GUG	Valine	GCG	Alanine	GAG	Glutamic Acid	GGG	Glycine	G	

3.4.2. What Is the Role of RNA in Protein Synthesis

3.4.2.1 Transcription Produces mRNA Molecules That Are Complementary Copies of One Strand of DNA

- The transcription of DNA into mRNA is restricted in two major ways.
  1. In any cell, transcription normally copies the DNA of only selected genes into mRNA.
  2. Transcription normally copies only one of the two strands of DNA into mRNA.
- In most cases, the useful information only reside on only one strand of the DNA double helix, cause the two strands of DNA are complementary not identical.
- The sequence bases of that strand codes for sequence of amino acids that forms the functional protein.
- The DNA strand that contains the gene and is transcribed into mRNA is called the template strand from which the complementary RNA strand is made.

Transcription consist of three process

1. Initiation
2. Elongation
3. Termination

Initiation :RNA Synthesis Begin at the Promoter of a Gene

- The synthesis of all three types of RNA is carried out by an enzyme called RNA polymerase.
- This enzyme uses genes on DNA as templates to specify the sequence of RNA nucleotides.
- RNA polymerase must locate the beginning of the gene for transcription to begin
- The promoter region of a gene is short sequence of DNA bases that marks the beginning of the gene.
- RNA polymerase binds to the DNA at the promoter site, initiating.

Elongation and Termination: RNA Synthesis Proceeds from the Promoter to the End of the Gene

- Once the RNA polymerase has bound to the promoter site, the enzyme changes shape, forcing the DNA double helix to unwind at the beginning of the gene.

- RNA polymerase then travels along the template strand, much as DNA polymerase does.
- Using free RNA nucleotides present in the nucleus, RNA polymerase synthesizes a single strand of RNA that is complementary to the template strand of DNA.
- After about 10 nucleotides have been added to the growing RNA chain, the beginning of the RNA molecule separate from the DNA and pairing between the bases on either DNA strand is re-established.
- As the RNA continues to elongate, it forms a long "tail" that drifts away from the DNA .
- RNA polymerase continues along the template strand until it reaches the termination signal, a sequence of DNA bases that triggers two events.
- First, the RNA molecules separates from both the DNA and the RNA polymerase.
- Second, the RNA polymerase detaches from the DNA template strands.

3.4.2.2 Messenger RNA Conveys the Code for Protein Synthesis from the Nucleus to the Cytoplasm

- mRNA carries the code for the amino acid sequence of a protein.
- In eukaryotic cells, mRNA molecules are synthesized in the nucleus and enter the cytoplasm through the pores in the nuclear envelope.
- In the cytoplasm, mRNA binds to ribosomes, where the codons of mRNA are translated into the amino acids in proteins.

3.4.2.3 Ribosomal RNA Forms an Important Part of the Protein-Synthesizing Machinery of a Ribosome.

- Ribosomes are composites of rRNA and many different protein.
- A single eukaryotic cell have tens of thousands of ribosome.
- Each ribosome is composed of two sub-units-one small and one large.
- The small sub-unit recognizes and binds mRNA and part of tRNA.
- The large ribosomal sub-unit contains an enzymatic region that catalyzes the addition of amino acids to growing protein chain, and it bears two other sites that bind to tRNA.

- Ribosomal RNA plays the major part in recognizing mRNA and in catalyzing the formation of peptide bonds between the amino acids that form the growing protein.

3.4.2.4 *Transfer RNA Molecules Decode the Sequence of Bases in mRNA into the Amino Acid Sequence of a Protein*

- Transfer RNA molecules binds to free amino acids in the cytoplasm and deliver them to ribosome, where, according to instructions from mRNA, they are incorporated into protein chains.
- There many types of tRNA molecules, at least one type for each amino acid.
- tRNA decipher the codons in mRNA and can translate them into an amino acid sequence.
- Enzyme in cytoplasm recognize each specific tRNA molecule and attach the correct amino acid to the molecule's stem.
- Energy of adenosine triphosphate (ATP) is stored in tRNA-amino acid bond.
- This energy is used later to forge a peptide bond when the amino acid is added to a growing protein molecule.
- tRNA bears three exposed bases, called the anticodon, that deciopher the code of the mRNA codon.
- Anticodon bases of each tRNA pair in a complementary manner to mRNA codon bases that specify the amino acid to which that tRNA can attach.

3.4.2.5 *Translation: mRNA, tRNA, and Ribosomes Cooperate to Synthesize Protein*

Translation consist of three steps:

1. Initiation of protein synthesis
2. Elongation of protein chain
3. Termination

Initiation: Protein Synthesis Begins When tRNA and mRNA Bind to a Ribosome

- Protein synthesis begins at specific start codon (AUG on mRNA).
- Translation begin by binding of several protein "initiation factors" and an initiator tRNA that bears the complementary "start anticodon" to small subunit ribosome.

- The small subunit then binds to an mRNA molecule and moves along it until the start codon is encountered.
- At this point, the tRNA base pairs with the mRNA start codon.
- The large ribosomal subunit then attaches to the small subunit.
- Simultaneously, the initiator tRNA binds to first binding site on the large subunit.
- The ribosome is now fully assembled and ready to begin translation.

Elongation and Termination: Protein Synthesis Proceeds One Amino Acids at a Time Until a Stop Codon Is Reached

- The anticodon of the a second tRNA, carrying another amino acid, recognizes the second mRNA codon and moves into the second binding site on the large subunit.
- The two amino acids carried by the two tRNA are now next to one another.
- The large subunit has a catalytic site that breaks the bond holding the first amino acid to its tRNA and forms a peptide bond between the two adjacent amino acids.
- Now, the first tRNA is empty whereas the second tRNA bears a two-amino-acid chain.
- The empty tRNA then drops off the ribosome and shift to the next codon on the mRNA molecule.
- The tRNA holding the elongating chain of amino acids shifts too, from the second to the first binding site of ribosome.
- A new tRNA, carrying another amino acid, binds to the empty second site.
- The catalytic site on the large subunit now links the third amino acids onto the growing chain.
- The empty tRNA leaves the ribosome, the ribosome shifts to another codon and the process repeats.
- Near the end of mRNA, a stop codon is reached and at this point a special enzymes cut the finished protein chain off the last tRNA, releasing the chain from the ribosome.

Chapter 4

SYSTEM OVERVIEW

This chapter gives an overview of the intelligent system developed under the GENETIC BIOLOGY presentation by describing their functions and capabilities. The design of the overall system is based on the framework of genetic biology analysis as discussed in Chapter 3. This is summarized in Figure 4.1.

The first system is known as GENSYS. This system represents a cell model that present the content of a cell and functions of each contents. The DNASYS is the second system that represents component of DNA and the construction of DNA. The last system is GENSYS, an intelligent system that represent the gene expression and regulation and the protein construction process in a cell. Combination of all this three system must give a fundamental instruction of genetic and protein construction in a living cell to the student or any personal.

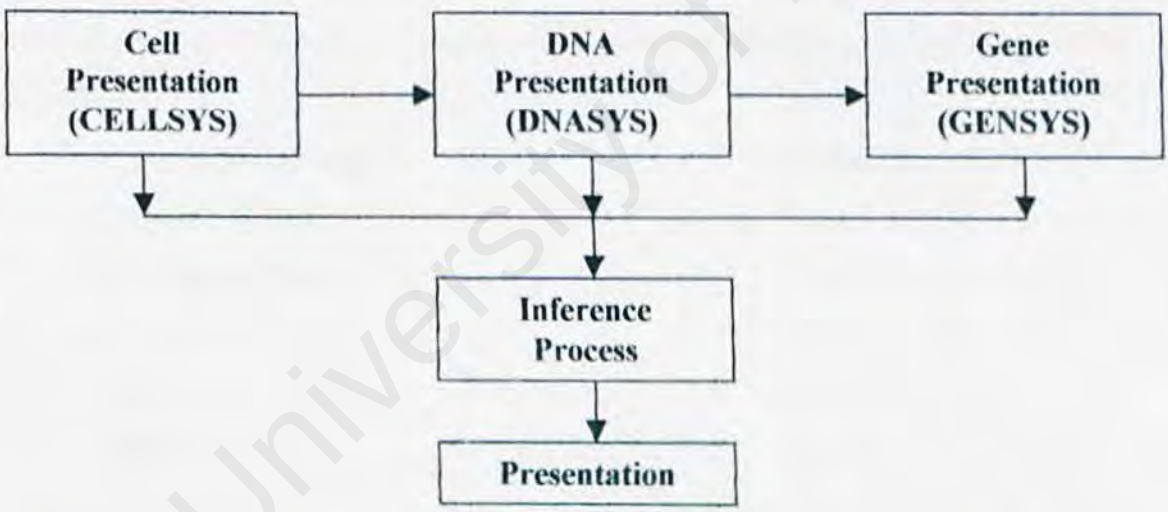

Figure 4.1 Instruction situation diagram for Genetic Biology presentation and the intelligent systems of GENETIC BIOLOGY project

The rest of this chapter is organized as follows. Section 4.1 states the general objectives of the GENETIC BIOLOGY project. Section 4.2 presents the classification of the GENETIC BIOLOGY systems. Section 4.3 presents the organization of the GENETIC BIOLOGY systems. Section 4.4 described the knowledge acquisition techniques used in the construction of the systems and Section 4.5 highlights the knowledge representation techniques employed.

4.1 OBJECTIVES OF THE GENETIC BIOLOGY PROJECT

The objectives of the research are broadly presented in Section 1.1. These objectives are summarized below:

- the presentation of genetic biology to strengthen the reliability of the instructions.
- The application of various knowledge representation techniques for enhancing the efficiency of the inference engine.
- to capture expertise in genetic biology presentation

4.2 CLASSIFICATION OF GENETIC BIOLOGY SYSTEMS

Hayes-Roth et al. Offer the following classification of intelligent system [Hayes-Roth et al., 1984]:

* Interpretation systems
* Diagnosis systems
* Debugging systems
* Debugging systems
* Instruction systems
* Text-based systems
* Prediction systems
* Design systems
* Monitoring systems
* Repair systems
* Control systems
* Dialogue-based systems

It must be noted that the above classification is not mutually exclusive as argued by Reichgelt & Van Harmelen [Reichgelt et al., 1986]. This is true when classifying the system of GENETIC BIOLOGY.

The systems of GENETIC BIOLOGY are basically dialogue-based question-answering system. Dialogue-based system diagnose, where natural language is used to query a database.

4.3 THE ORGANIZATION OF THE INTELLIGENT'S SYSTEM

In this section, an overview of the organization of the GENETIC BIOLOGY system is presented. Chapter 2 has illustrated the basic components of an intelligent system. A general organization of an GENETIC BIOLOGY system is presented in Figure 4.2.

The basic components of GENETIC BIOLOGY are described below:

1. An interactive module supporting an interactive dialogue between the system and the user. This module translates requests and replies from an external into an internal language, and to manage the dialogue
2. A problem-solving module, which acts as an inference mechanism, making it possible to generate possible solution during the reasoning process, and as a reasoning mechanism, permitting efficient operation of the inference mechanism.

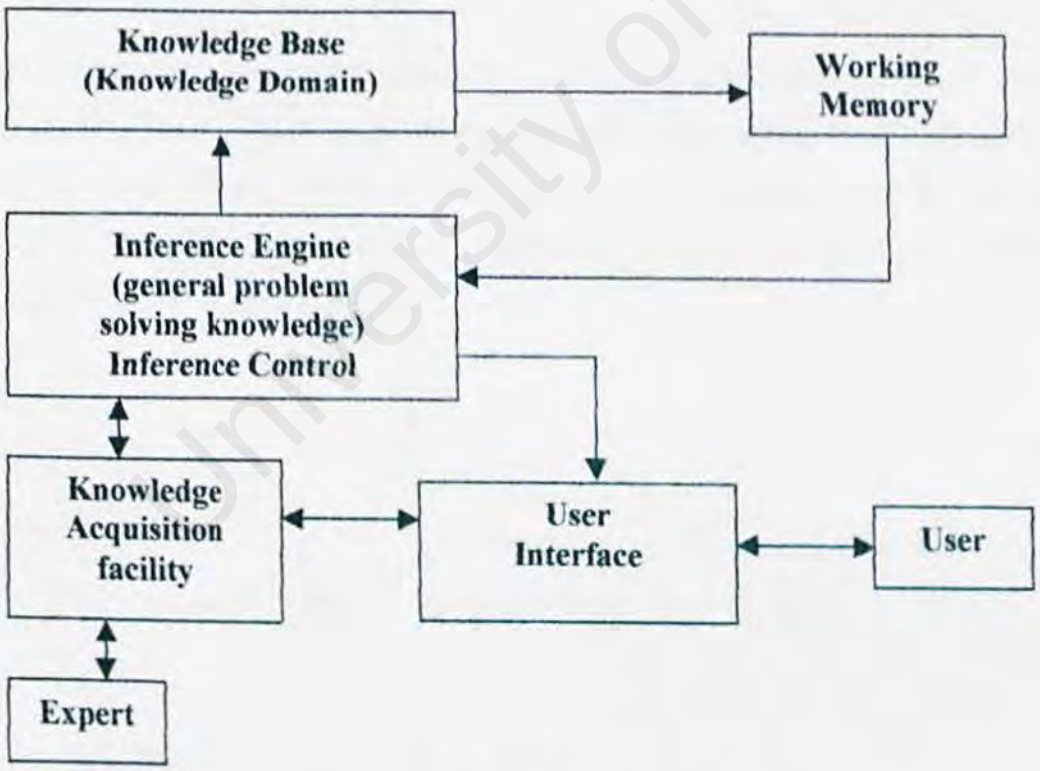

Figure 4.2 The architecture of an GENETIC BIOLOGY system

4.4 KNOWLEDGE ACQUISITION TECHNIQUES

Buchanan et al. Defines knowledge acquisition as a process model of how to construct an intelligent system [Buchanan et al., 1984]. Figure 4.3 summarizes the stages of knowledge acquisition in the GENETIC BIOLOGY project.

The first task in knowledge acquisition is to set up meetings with the relevant experts to establish a precise system of methods in genetic analysis. Two experts were involved in the project. One of them is an expert in genetic. The other is a biology teacher who face the user (student) problem in understanding genetic topics.

The process by which the knowledge engineer acquires the knowledge from the expert is known as knowledge engineering. This process can be divided into two stages: the discovery of "public" knowledge, and the discovery of "private" knowledge [Hayes-Roth et al., 1984]. The construction of GENETIC BIOLOGY requires the acquisition of the public (mostly) and private knowledge.

Public knowledge is knowledge that can be gained from public sources such as textbooks, journals or articles. The knowledge engineer can easily obtain these materials from a relatively good library. Most of these materials are recommended by the experts. GENETIC BIOLOGY uses a number of references such as Aurerisk Teresa [1999],World Scientific [1994], Kleinsmith [1995].

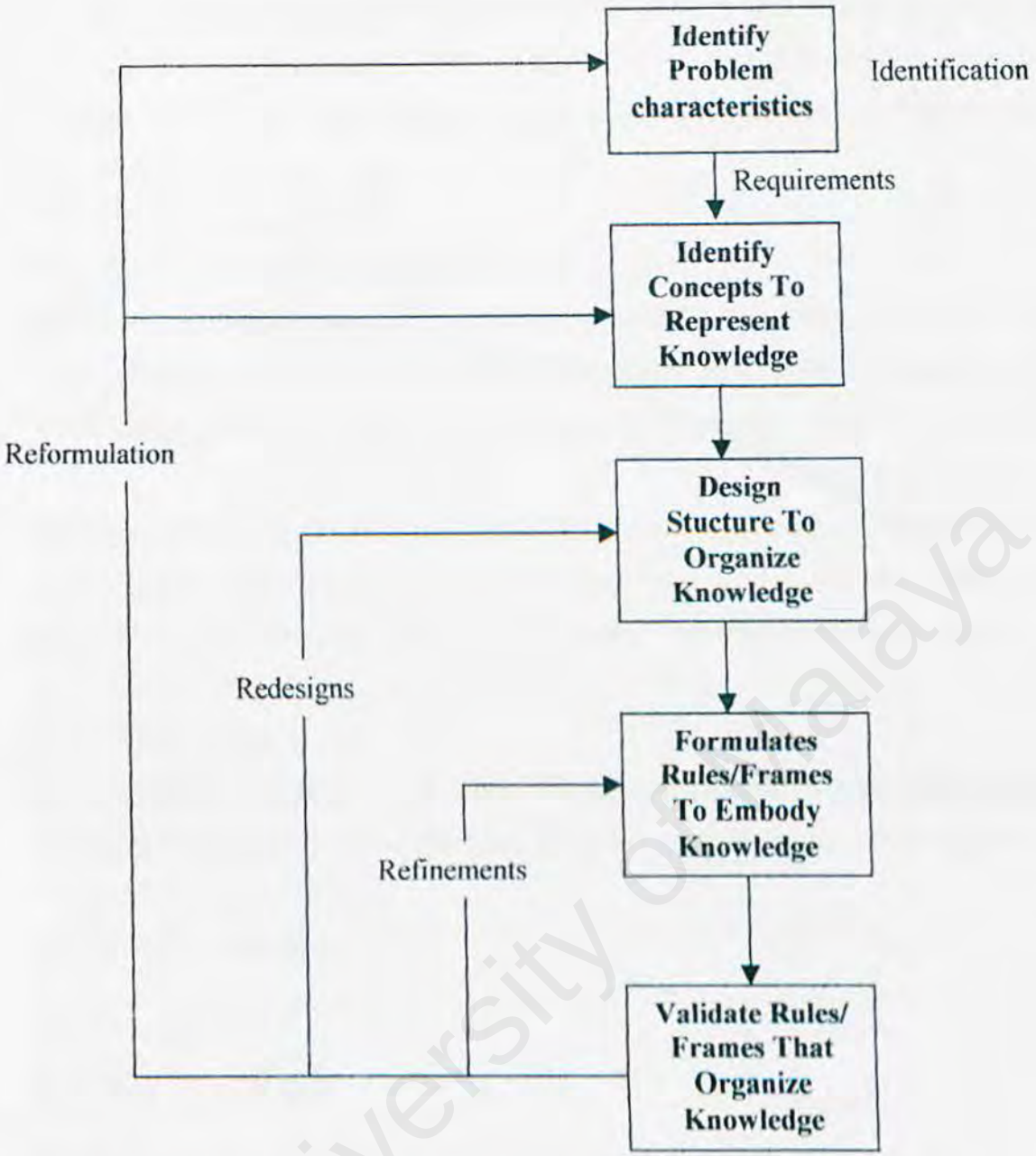

Figure 4.3 Stages of knowledge acquisition in GENETIC BIOLOGY

To make the intelligent system truly "intelligent", the knowledge engineer obtains private knowledge from the expert. This type of knowledge is procedural and declarative in nature which the expert possesses and applies during real analysis. GENETIC BIOLOGY never employs heuristic knowledge for this purpose because this system is based on a well known facts and processes.

The process of acquiring private knowledge is an iterative process that requires close interaction between these experts and the knowledge engineer. It is found to be easier to incorporate the public knowledge first in rough system and subsequently demonstrate the system to the experts for refinements to be made.

4.5 KNOWLEDGE REPRESENTATION

The expert's knowledge can be thought of as both passive and active knowledge. A fact an expect knows to be true is considered passive knowledge, while a method an expert uses to take deductions or inferences about the subject is considered active knowledge.

The knowledge representation of GENETIC BIOLOGY can be classified into two broad groups, namely micro and macro levels. At macro level, a block of related knowledge is represented using frames. At micro level, individual knowledge is treated using rules.

4.5.1 Frame-based

This approach is appropriate if the expert describes the problem by referencing important objects and their relationships. The general structure of this representation is known in Figure 4.4.

CONCEPT (identifier)	
SLOT (property) I	Value I
SLOT (property) II	Value I
SLOT (property) III	Value I

Figure 4.4 A simple frame structure

The above frame structure represents the identifier-property- value structure. It is a network of nodes and relations organized in a hierarchy, where the topmost nodes represent general concepts and the lower nodes more specific instances of those concepts [Waterman, 1986]. A concept serves as identifier to a frame. A slot holds the property of

the frame. This techniques is used in representing the CELLSYS module and part of the DNASYS module GENSYS module.

4.5.2 Rule-based

Rules are operational granules of knowledge. Thus, they are used at micro level and are generally expressed as IF- THEN statements, as shown below.

IF UUU THEN Phenylalanine

Each rule is identified by a name. Following the name is the IF part of the rule. The section of the rule between the IF and THEN part of the rule is called by various names such as the antecedent, conditional part, pattern part or left-hand-side (LHS). Following the THEN part of the rule is a list of actions to be executed when the rule fires. This part of the rule is known as the consequent or right-hand-side (RHS). This approach is suitable if the problem is to be identified in a procedural manner. This techniques is used in representing the DNA production steps in DNASYS and gene expression and regulation in GENSYS.

4.5.3 Control Techniques

As discussed in earlier chapter, only forward chaining will be used in the problem inference technique. This is because there is no well define words or symbols for genetic, to employ backward chaining. Given a fact(entity) through the query, the subject must be represented with the solution.

4.6 AN OVERVIEW OF GENETIC BIOLOGY KNOWLEDGE BASE

4.6.1 CELLSYS Knowledge Base

Class Name:	Cell	
Subclasses:	Prokaryotic, Eukaryotic	
Properties:	Cell Surface	Present
	Nucleus	Present
	Cytoplasm	Present

Figure 4.5 General Cell Class Frame

Class Name:	Eukaryotic	
Subclasses:	Plants, Animals	
Properties:	Cell Surface	Present
	Nucleus	Present
	Cytoplasm	Present

Figure 4.6 Eukaryotic Cell Class Frame

Class Name: Prokaryotic

Subclasses: Bacteria, Archaeans

Properties:

Cell Wall	Present
Plasma membrane	Present
Genetic material	DNA
Chromosomes	Single, circular, no protein
Nucleus	Absent
Nuclear envelope	Absent
Nucleolus	Absent
Mitochondria	Absent
Chloroplasts	Absent
Ribosomes	Present
Endoplasmic reticulum	Absent
Golgi complex	Absent
Lysosomes	Absent
Plastids	Absent
Central vacuole	Absent
Other vesicles and vacuoles	Absent
Cytoskeleton	Absent
Centrioles	Absent
Cilia and flagella	Present

Figure 4.7 Prokaryotic Class Frame

Class Name: Plant Cell

Subclasses:

Properties:

Cell Wall	Present
Plasma membrane	Present
Genetic material	DNA
Chromosomes	Many, linear, with protein
Nucleus	Present
Nuclear envelope	Present
Nucleolus	Present
Mitochondria	Present
Chloroplasts	Present
Ribosomes	Present
Endoplasmic reticulum	Present
Golgi complex	Present
Lysosomes	Present
Plastids	Present
Central vacuole	Present
Other vesicles and vacuoles	Present(some)
Cytoskeleton	Present
Centrioles	Absent(in most)
Cilia and flagella	Absent(in most)

Figure 4.8 Plant Cell Class Frame

Class Name: Animal Cell

Subclasses:

Properties:

Cell Wall	Absent
Plasma membrane	Present
Genetic material	DNA
Chromosomes	Many, linear, with protein
Nucleus	Present
Nuclear envelope	Present
Nucleolus	Present
Mitochondria	Present
Chloroplasts	absent
Ribosomes	Present
Endoplasmic reticulum	Present
Golgi complex	Present
Lysosomes	Present
Plastids	Absent
Central vacuole	Absent
Other vesicles and vacuoles	Present(some)
Cytoskeleton	Present
Centrioles	Present
Cilia and flagella	Present

Figure 4.9 Animal Cell Class Frame

Frame Name: Cell

Properties:

Cell Surface	Outer Construction
Nucleus	Organize genetic material
Cytoplasm	Consists water, salts and organic molecules

Figure 4.10 General Cell Function Frame

Frame Name: Cell Surface

Class: Cell

Properties:

Cell Wall	- Protects - Supports cell
Plasma membrane	- Isolates cell content from environment - Regulates movement of material into and out of cell - Communicates with other cells

Figure 4.11 Cell Surface Frame

Frame Name:	Nucleus
Class:	Cell

Properties:

DNA	- Encodes information needed to construct cell and control cellular activity
Chromosomes	- Contain and control use of DNA
Nucleus	- Membrane-bound container for Chromosomes
Nuclear	- Synthesizes ribosomes

Figure 4.12 Cell Nucleus Frame

Frame Name:	Cytoplasma
Class:	Cell

Properties:

Mitochondria	- Produce energy by aerobic metabolism
Chloroplasts	- Perform photosynthesis
Ribosomes	- Provide site of protein synthesis
Endoplasmic reticulum	- Synthesizes membrane components linids

Golgi complex	- Modifies and packages proteins and lipids; - Synthesizes carbohydrates
Lysosomes	- Contain intracellular digestive enzymes
Plastids	- Store food, pigments
Central vacuole	- Contains water and wastes - Provides turgor pressure to support cell
Other vesicles and vacuoles	- Contains food obtained through phagocytosis; - Contains secretory product
Cytoskeleton	- Gives shape and support to cell - Positions and moves cell parts
Centrioles	- Synthesize microtubules of cilia and flagella - May produce spindle in animal cells
Cilia and flagella	- Move cell through fluid or move Fluid past cell surface

Figure 4.13 Cell Cytoplasma Frame

4.6.2 DNASYS Knowledge Base

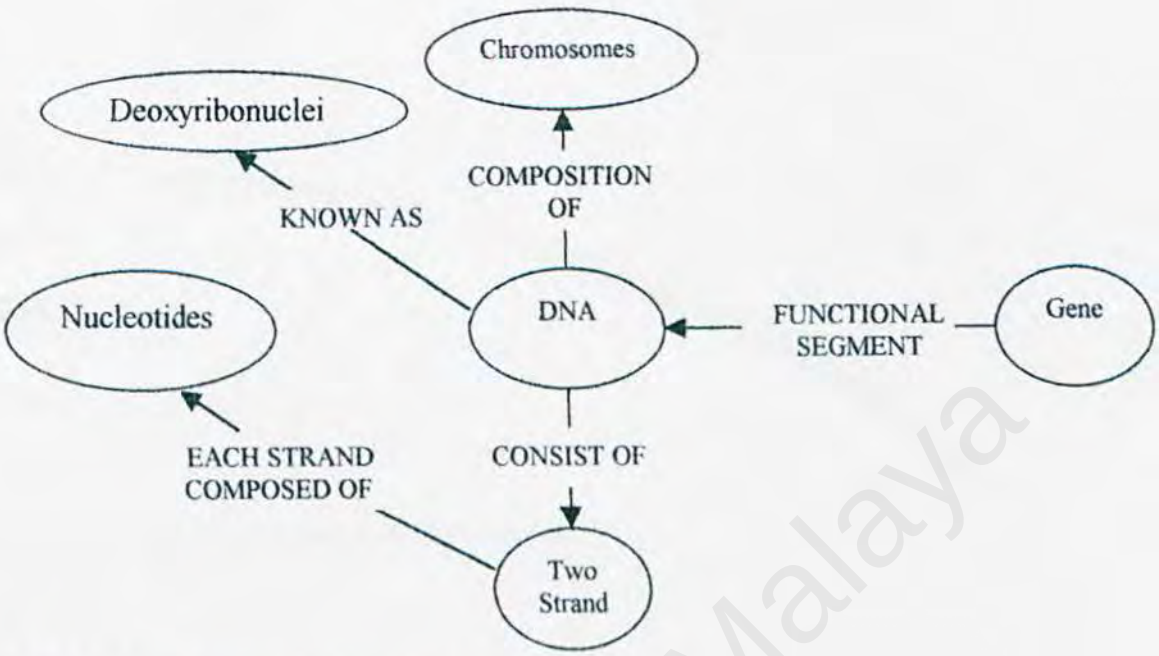

Figure 4.14 Basic Semantic Network of DNA

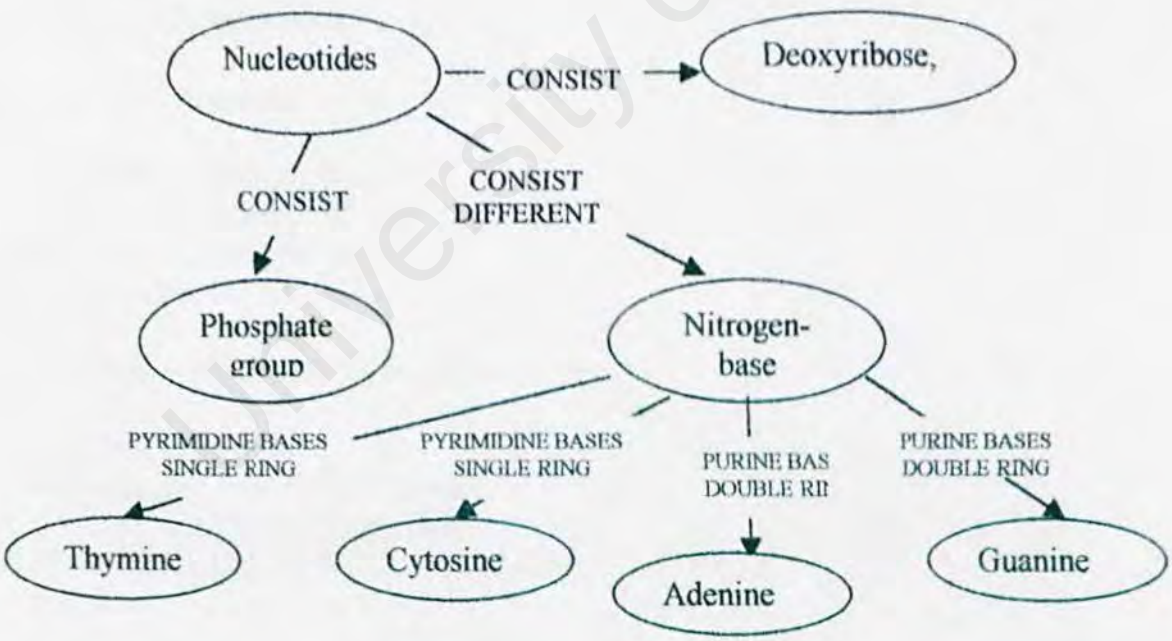

Figure 4.15 Semantic Network of Nucleotides

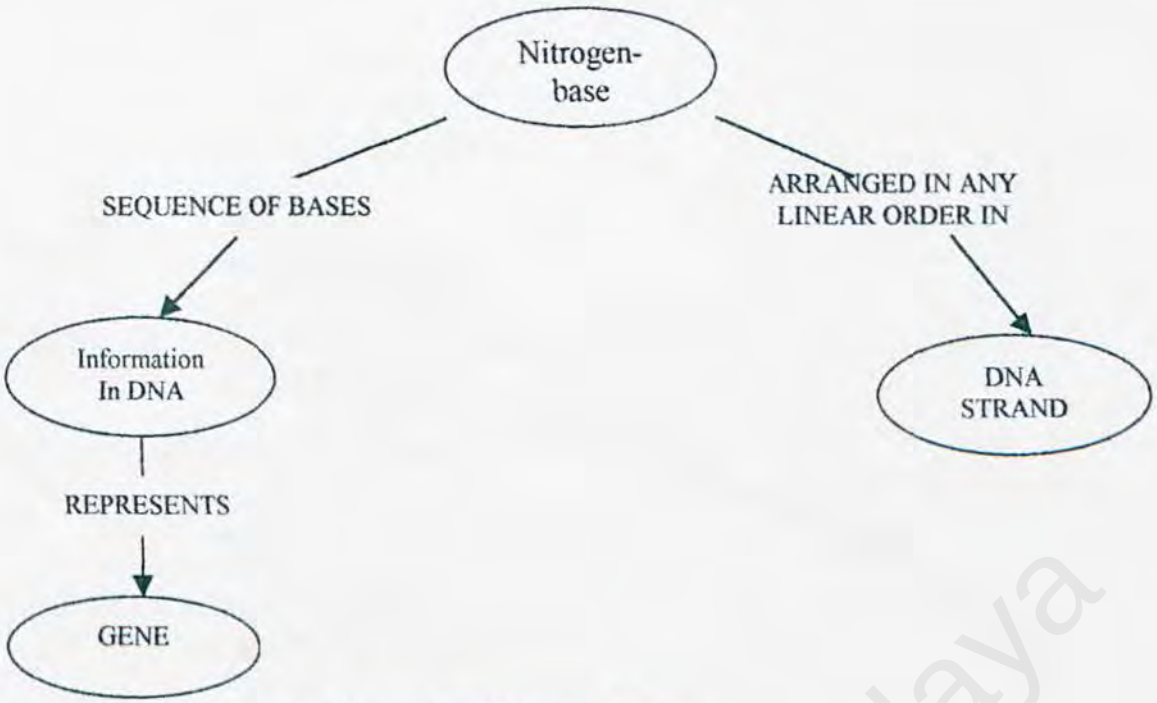

Figure 4.16 Semantic Network of Nitrogen-Base

DNA Nucleotides Base Rules

Goal DNA Nucleotide Base?

RULE 1DNA Nucleotide

IF Consist Nitrogen Base
AND Consist Phosphate Group
AND Consist Deoxyribose
THEN DNANucleotide

RULE 2 DNA Nucleotide Thymine Base

IF Base is Thymine
AND Nitrogen Base is Pyrimidine
AND Nitrogen Base is Single Ring
THEN DNA Nucleotide is Thymine Base

RULE 3 DNA Nucleotide Cytosine Base

IF Base is Cytosine
AND Nitrogen Base is Pyrimidine
AND Nitrogen Base is Single Ring
THEN DNA Nucleotide is Cytosine Base

RULE 4 DNA Nucleotide Adenine Base

IF Base is Adenine
AND Nitrogen Base is Purine
AND Nitrogen Base is Double Ring
THEN DNA Nucleotide is Adenine Base

RULE 5 DNA Nucleotide Guanine Base

IF Base is Guanine
AND Nitrogen Base is Purine
AND Nitrogen Base is Double Ring
THEN DNA Nucleotide is Guanine Base

Figure 4.17 DNA Nucleotides Base Rules

DNA Structure Rules

Goal Structure is DNA?

RULE 1 DNA Structure

IF Consist of Two Strand
AND Two Strand are Twisted into Double Helix
AND Double Helix Shape like Corkscrew Ladder
AND Each Strand Composed of Nucleotides
AND Each Nucleotides Bonded to Each Other
AND Bonding Form Backbone
AND Backbone of Two Strand(Double Helix) Face Outside
THEN Structure is DNA

RULE 2 Double Helix

IF Hidrogen Bonds Complementary Base Pairs
AND Bonds in Center of Double Helix
THEN Strand in Double Helix

RULE 3 Complementary Base Pair

IF Guanine and Cytosine
OR Adenine and Thymine
THEN Complementary Base Pair

RULE 4 Composed of Nucleotides

IF Series of Nucleotides
AND Has Free Sugar End at One Side of Strand
AND Has Free Phosphate End at Other Side of Strand
THEN Each Strand Composed of Nucleotides

RULE 5 Nucleotides Bonded

IF Phosphate Group and Sugar Bonded
AND Phosphate Group and Sugar of Different Nucleotides
AND Phosphate Group and Sugar Arranged Alternatively
THEN Each Nucleotides Bonded to Each Other

RULE 6 Backbone

IF Bonding Form Long Strand of Nucleotides
AND Bases Protruding From Nucleotides
THEN Bonding Form Backbone

Figure 4.18 DNA Structure_Rules

DNA Semiconservative Replication Rules

Goal DNA Semiconservative Replication?

RULE 1 DNA Replicate

IF Two DNA Strands is Separated
AND New Strands is Synthesis
AND Produce Two Identical DNA Double Helices
THEN DNA Replicate

RULE 2 DNA Strands is Separated

IF Parental DNA Strands Unwind
AND Hydrogen Bonds Between Complementary Bases Broken
THEN Two DNA Strands is Separated

RULE 3 New Strands is Synthesis

IF DNA Strands is Separated
AND Daughter Strands Produce
THEN New Strands is Synthesis

RULE 4 Produce Two Identical DNA Double Helices

IF New Strands is Synthesis
AND Chromosome Consist One Parental Strand And One Complementary Daughter Strand
THEN Produce Two Identical DNA Double Helices

RULE 5 Produce Daughter Strands

IF Complementary Nucleotides is Added to Growing Daughter Strand
THEN Daughter Strands Produce

Figure 4.19 DNA Semiconservative Replication Rules

Parental DNA Strands Unwind Rules

Goal Parental DNA Unwind?

RULE 1 Parental DNA Unwind

IF Enzyme DNA Helicase Break All Hidrogen Bonds
AND Nucleotide Bases is Exposed
THEN Parental DNA Unwind

RULE 2 All Hidrogen Bonds Broken

IF DNA Helicase Past Through Double Helix Bonds
THEN DNA Helicase Break All Hidrogen Bonds

Figure 4.20 Parental DNA Unwinding Rules

Complementary Daughter DNA Strands Synthesization Rules

Goal Daughter Strand Synthesized?

RULE 1 Daughter Strand Synthesized

IF Enzyme DNA Polymerase Binds to Unwound DNA Strands
AND One DNA Polymerase to Each Parental Strands
AND DNA Polymerase Identify Bases Exposed in Parental Strands
AND DNA Polymerase Match Exposed Bases to Free Complementary Nucleotides
AND DNA Polymerase Forms Daughter Strand Backbone
AND DNA Polymerase Travel One Direction
AND Both DNA Polymerase in Opposite Direction
AND DNA Polymerase Follows Hidrogen Bond Separating DNA Helicase Forms One Daughter Strand
AND DNA Polymerase Move Away From Hidrogen Bond Separating DNA Helicase Forms Pieces of Strand
AND New DNA Polymerase
AND Pieces of Strands Bonded Together
THEN Daughter Strand Synthesized

RULE 2 Match Is Made

IF Adenine Base Nucleotide Forms Hydrogen Bonds With Exposed
OR Thymine Base Nucleotide Forms Hydrogen Bonds With Exposed Adenine
OR Guanine Base Nucleotide Forms Hydrogen Bonds With Exposed Cytosine
OR Cytosine Base Nucleotide Forms Hydrogen Bonds With Exposed Guanine
THEN Match Exposed Bases to Free Complementary Nucleotides

RULE 3 Daughter Strands Form Backbone

IF DNA Polymerase Bonds Together The Sugars And Phosphates of Complementary Nucleotides
THEN DNA Polymerase Forms Daughter Strand Backbone

RULE 4 DNA Polymerase Travel One Direction

IF DNA Polymerase "Free Sugar" End To The "Free Phosphate" End
OR DNA Polymerase "Free Phosphate" End To The "Free Sugar" End
THEN DNA Polymerase Travel One Direction

RULE 5 New DNA Polymerase

IF The Piece of Strand is of DNA Polymerase Which Moving Away From DNA Helicase
THEN New DNA Polymerase

RULE 6 Bond Two Strands

IF The Daughter Strand is of DNA Polymerase Which Moving Away From DNA Helicase
AND New DNA Polymerase
AND New Strands Synthesized
AND Enzyme DNA Ligase Bonds The Two Piece of Strands
AND This Bonding Repeated For All Strand Synthesized
THEN Pieces of Strands Bonded Together

Figure 4.21 Complementary Daughter DNA Strands Synthesization Rules

4.6.3 GENSYS Knowledge Base

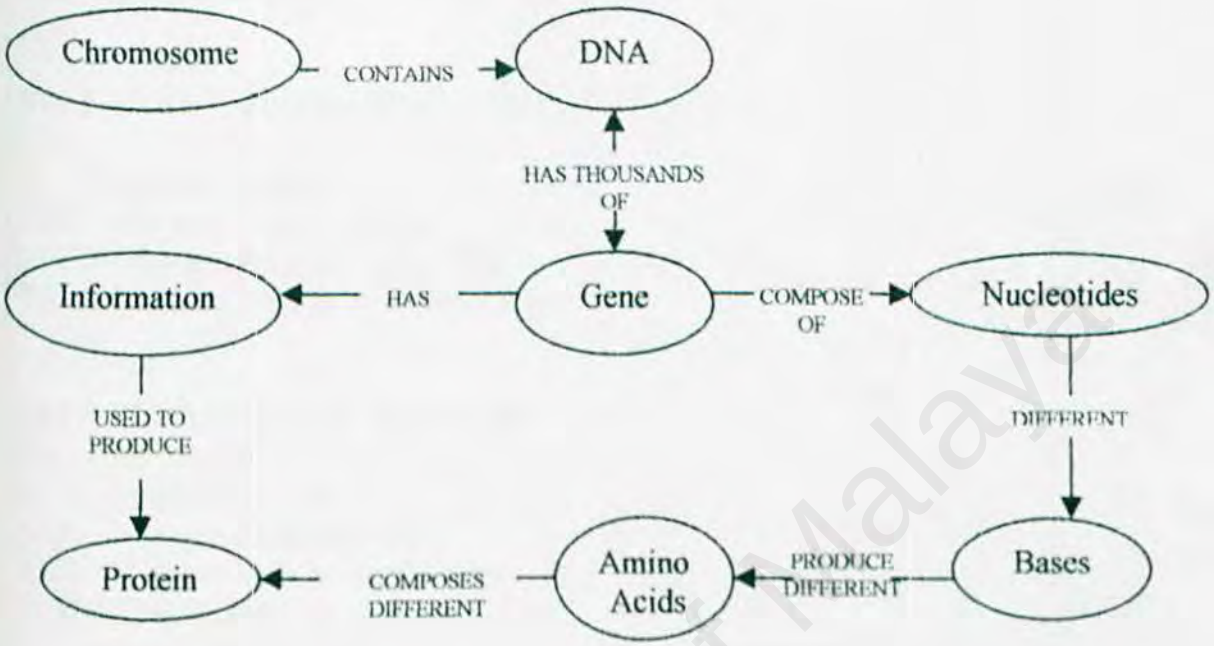

Figure 4.22 Semantic Network of Gene

RNA Nucleotides Base Rules

```
Goal RNA Nucleotide Base?

RULE 1RNA Nucleotide

IF      Consist Nitrogen Base
AND     Consist Phosphate Group
AND     Consist Ribose
THEN RNA Nucleotide

RULE 2 RNA Nucleotide Uracil Base

IF      Base is Uracil
AND     Nitrogen Base is Pyrimidine
AND     Nitrogen Base is Single Ring
THEN RNA Nucleotide  is Uracil Base
```

RULE 3 RNA Nucleotide Cytosine Base

IF Base is Cytosine
AND Nitrogen Base is Pyrimidine
AND Nitrogen Base is Single Ring
THEN RNA Nucleotide is Cytosine Base

RULE 4 RNA Nucleotide Adenine Base

IF Base is Adenine
AND Nitrogen Base is Purine
AND Nitrogen Base is Double Ring
THEN RNA Nucleotide is Adenine Base

RULE 5 RNA Nucleotide Guanine Base

IF Base is Guanine
AND Nitrogen Base is Purine
AND Nitrogen Base is Double Ring
THEN RNA Nucleotide is Guanine Base

Figure 4.23 RNA Nucleotides Base Rules

RNA Types Rules

Goal Type of RNA?

RULE 1 mRNA

IF Carries The Code From Genes To Ribosomes
AND Has Thousands of Bases Complementary to DNA Strand
AND Has Start Codon
AND Has Start Codon
THEN mRNA
OR Messenger RNA

RULE 2 rRNA

IF Combines With Protein To Form Ribosomes
THEN rRNA

```
OR      Ribosomal RNA


RULE 3 tRNA

IF      Carries Amino Acids To Ribosome
THEN tRNA
OR      Tranfer RNA
```

Figure 4.24 RNA Types Rules

Start Codon Rules

```
Goal    Start Codon?

RULE Start Codon

IF      Base Sequence is Adenine, Uracil , Guanine (AUG)
THEN Start Codon
```

Figure 4.25 Start Codon Rules

Stop Codon Rules

```
Goal    Stop Codon?

RULE Stop Codon

IF      Base Sequence is Uracil , Adenine, Guanine (UAG)
OR      Base Sequence is Uracil , Adenine, Adenine (UAA)
OR      Base Sequence is Uracil , Guanine, Adenine (UGA)
THEN Start Codon
```

Figure 4.26 Stop Codon Rules

Ribosome Structure Rules

```
Goal   Ribosome Structure?

RULE Ribosome Structure

IF     Composites Of rRNA
AND  Has One Small Sub-Unit
AND  Has One Large Sub-Unit
AND  Large Sub-Unit Has Two Binding Site
AND  Large Sub-Unit Has One Catalytic Site
THEN Ribosome Structure
```

Figure 4.27 Ribosome Structure Rules

Protein Synthesis Rules

```
Goal Protein Synthesis?

RULE Protein Synthesis

IF     Transcription
AND  Messenger From The Nucleus To The Cytoplasm
AND  Ribosomes In Cytoplasm
AND  Translation
THEN Protein Synthesized
```

Figure 4.28 Protein Synthesis Rules

Transcription Rules

```
Goal   Transcription?

RULE 1 Transcription


IF     Initiation Messenger RNA
AND  Elongation Messenger RNA
AND  Termination Messenger RNA
```

THEN Transcription

RULE 2 Initiation Messenger RNA

IF RNA Polymerase Locate Beginning of The Gene for Transcription
AND Location is Called Promoter Region
AND Promoter Region is Sequences of DNA Bases
AND Promoter Region Marks The Beginning of Gene
AND RNA Polymerase Binds to The DNA at The Promoter Site
AND RNA Polymerase Begin Transcripting DNA Bases
THEN Initiation Messenger RNA

RULE 3 Elongation Messenger RNA

IF RNA Polymerase Changes Shape Beginning of The Gene
AND RNA Polymerase Force The DNA Double Helix to Unwind
AND RNA Polymerase Travels Along The Template Strand
AND RNA Nucleotides Produced
AND RNA Strands Produced
THEN Elongation Messenger RNA

RULE 4 RNA Nucleotides Produced

IF Using Free RNA Nucleotides Present in The Nucleus
AND RNA Polymerase Synthesizes RNA Nucleotides
THEN RNA Nucleotides Produced

RULE 5 RNA Strands Produced

IF After About 10 Nucleotides Have Been Added To The Growing RNA Chain
AND Beginning of the RNA Molecule Separate From The DNA
AND Pairing Between The Bases On Either DNA Strand Is Re-Established
AND RNA Continues To Elongate
AND RNA Drifts Away From The DNA
AND RNA Polymerase Reaches The Termination Signal
AND Strand Of RNA Complementary To The Template Strand Of DNA
THEN RNA Strands Produced

RULE 6 Termination Messenger RNA

IF RNA Molecules Separates From Both The DNA And The RNA Polymerase
AND RNA Polymerase Detaches From The DNA Template Strands
THEN Termination Messenger RNA

Figure 4.29 Transcription Rules

Messenger RNA from the Nucleus to the Cytoplasm Rules

Goal Messenger RNA From The Nucleus To The Cytoplasm ?

RULE Messenger RNA From The Nucleus To The Cytoplasm

IF Messenger RNA Molecules Are Synthesized In The Nucleus
AND Messenger RNA Molecules Enter The Cytoplasm Through The Pores In The Nuclear Envelope
THEN Messenger RNA From The Nucleus To The Cytoplasm

Figure 4.30 Messenger RNA from the Nucleus to the Cytoplasm Rules

Translation Rules

Goal Translation?

RULE 1 Translation

IF Initiation Of Protein Synthesis
AND Elongation Of Protein Chain
AND Termination Protein Synthesis
THEN Translation

RULE 2 Initiation Of Protein Synthesis

IF Messenger RNA Binds To Ribosomes Small Sub-Unit
AND Small Sub-Unit Moves Along Messenger RNA Until Start Codon Is Encountered
AND Transfer RNA Binds To First Site Of Transfer RNA Binding Site
AND Transfer RNA Base Pairs With The Messenger RNA Start Codon
AND Protein Synthesis Begins At Complementary Start AntiCodon (on Transfer RNA)
AND Ribosomal Large Subunit Attaches To The Small Subunit
THEN Initiation Of Protein Synthesis

RULE 3 Elongation Of Protein Chain

IF Catalytic Site Catalyzes Amino Acids
AND Anticodon Of The A Second Transfer RNA Carry Another Amino Acid

AND Recognizes The Second Messenger RNA Codon
AND Second Transfer RNA Binds To Second Site Of Transfer RNA Binding Site
AND Two Amino Acids Carried By The Two Transfer RNA Are Next To One Another
AND Catalytic Site Breaks The Bond Holding The First Amino Acid To Its Transfer RNA
AND Peptide Bond Formed Between The Two Adjacent Amino Acids
AND First Transfer RNA Is Empty
AND The Second Transfer RNA Bears a Two-Amino-Acid Chain
AND Empty Transfer RNA Drop Off
AND Second Transfer RNA Shifted To First Transfer Binding side
AND New Transfer RNA Carry Another Amino Acid
AND New Transfer RNA Binds To The Empty Second Site
AND Third Amino Acid Catalyzed Added To Growing Protein Chain
AND Protein Synthesis Proceeds One Amino Acids at a Time Until a Stop Codon Is Reached
THEN Elongation Of Protein Chain

RULE 3 Termination Protein Synthesis

IF Stop Codon Is Reached
AND Special Enzymes Cut The Finished Protein Chain Off The Last Transfer RNA
AND Protein Chain Release From Ribosome
THEN Termination Protein Synthesis

Figure 4.31 Translation Rules

Codon Rules

Goals Codon?

RULE Form One Codon

IF Three Bases
THEN Form One Codon

Figure 4.32 Codon Rules

Amino Acid Type Rules

Goal Amino Acid?

RULE 1 Phenylalanine

IF Codon Composed of Uracil, Uracil, Uracil (UUU)
OR Codon Composed of Uracil, Uracil, Cytosine (UUC)
THEN Phenylalanine

RULE 2 Leucine

IF Codon Composed of Uracil, Uracil, Adenine (UUA)
OR Codon Composed of Uracil, Uracil, Guanine (UUG)
OR Codon Composed of Cytosine, Uracil, Uracil (CUU)
OR Codon Composed of Cytosine, Uracil, Cytosine (CUC)
OR Codon Composed of Cytosine, Uracil, Adenine (CUA)
OR Codon Composed of Cytosine, Uracil, Guanine (CUG)
THEN Leucine

RULE 3 Isoleucine

IF Codon Composed of Adenine, Uracil, Uracil (AUU)
OR Codon Composed of Adenine, Uracil, Cytosine (AUC)
OR Codon Composed of Adenine, Uracil, Adenine (AUA)
THEN Isoleucine

RULE 4 Start(Methionine)

IF Codon Composed of Adenine, Uracil, Guanine (AUU)
THEN Start(Methionine)

RULE 5 Valine

IF Codon Composed of Guanine, Uracil, Uracil (GUU)
OR Codon Composed of of Guanine, Uracil, Cytosine (GUC)
OR Codon Composed of Guanine, Uracil, Adenine (GUA)
OR Codon Composed of Guanine, Uracil, Guanine (GUG)
THEN Valine

RULE 6 Serine

IF Codon Composed of Uracil, Cytosine, Uracil (UCU)
OR Codon Composed of of Uracil, Uracil, Cytosine (UUC)
OR Codon Composed of Uracil, Cytosine, Adenine (UCA)
OR Codon Composed of Uracil, Cytosine, Guanine (GUG)
OR Codon Composed of Adenine, Guanine, Uracil (AGU)
OR Codon Composed of Adenine, Guanine, Cytosine (AGC)
THEN Serine

RULE 7 Proline

IF Codon Composed of Cytosine, Cytosine, Uracil (CCU)
OR Codon Composed of Cytosine, Cytosine, Cytosine (CCC)
OR Codon Composed of Cytosine, Cytosine, Adenine (CCA)
OR Codon Composed of Cytosine, Cytosine, Guanine (CCG)
THEN Proline

RULE 8 Threonine

IF Codon Composed of Adenine, Cytosine, Uracil (ACU)
OR Codon Composed of Adenine, Cytosine, Cytosine (ACC)
OR Codon Composed of Adenine, Cytosine, Adenine (ACA)
OR Codon Composed of Adenine, Cytosine, Guanine (ACG)
THEN Threonine

RULE 9 Alanine

IF Codon Composed of Guanine, Cytosine, Uracil (GCU)
OR Codon Composed of Guanine, Cytosine, Cytosine (GCC)
OR Codon Composed of Guanine, Cytosine, Adenine (GCA)
OR Codon Composed of Guanine, Cytosine, Guanine (GCG)
THEN Alanine

RULE 10 Tyrosine

IF Codon Composed of Uracil, Adenine, Uracil (UAU)
OR Codon Composed of Uracil, Adenine, Cytosine (UAC)
THEN Tyrosine

RULE 11 Stop

IF Codon Composed of Uracil, Adenine, Adenine (UAA)
OR Codon Composed of Uracil, Adenine, Guanine (UAG)
OR Codon Composed of Uracil, Guanine, Adenine (UGA)
THEN Stop

RULE 12 Histidine

IF Codon Composed of Cytosine, Adenine, Uracil (CAU)
OR Codon Composed of Cytosine, Adenine, Cytosine (CAC)
THEN Histidine

RULE 13 Glutamine

IF Codon Composed of Cytosine, Adenine, Adenine (CAA)
OR Codon Composed of Cytosine, Adenine, Guanine (CAG)
THEN Glutamine

RULE 14 Asparagine

IF Codon Composed of Adenine, Adenine, Uracil (AAU)
OR Codon Composed of Adenine, Adenine, Cytosine (AAC)
THEN Asparagine

RULE 15 Lysine

IF Codon Composed of Adenine, Adenine, Adenine (AAA)
OR Codon Composed of Adenine, Adenine, Guanine (AAG)
THEN Lysine

RULE 16 Aspartic Acid

IF Codon Composed of Guanine, Adenine, Uracil (GAU)
OR Codon Composed of Guanine, Adenine, Cytosine(GAC)
THEN Aspartic Acid

RULE 17 Glumatic Acid

IF Codon Composed of Guanine, Adenine, Adenine (GAA)
OR Codon Composed of Guanine, Adenine, Guanine (GAG)
THEN Glumatic Acid

RULE 18 Cyteine

IF Codon Composed of Uracil, Guanine, Uracil (UGU)
OR Codon Composed of Uracil, Guanine, Cytosine (UGC)
THEN Cyteine

RULE 19 Tryptophan

IF Codon Composed of Uracil, Guanine, Guanine (UGG)
THEN Tryptophan

RULE 20 Arginine

IF Codon Composed of Cytosine, Guanine, Uracil (CGU)
OR Codon Composed of of Cytosine, Guanine, Cytosine (CGC)
OR Codon Composed of Cytosine, Guanine, Adenine (CGA)
OR Codon Composed of Cytosine, Guanine, Guanine (CGG)
OR Codon Composed of Adenine, Guanine, Adenine (AGA)
OR Codon Composed of Adenine, Guanine, Guanine (AGG)
THEN Arginine

RULE 21 Glycine

IF Codon Composed of Guanine, Guanine, Uracil (GGU)
OR Codon Composed of Guanine, Guanine, Cytosine (GGC)
OR Codon Composed of Guanine, Guanine, Adenine(GGA)
OR Codon Composed of Guanine, Guanine, Guanine (GGG)
THEN Glycine

Figure 4.33 Amino Acid Type Rules

CONCLUSIONS

The purpose of this thesis has been to describe the development of intelligent system for Genetic Biology tutoring. Through out this thesis we have focused on scope and objective of the system. We also have done literature review that covers major aspects of intelligent system technology in chapter 2. In chapter 3 a framework of genetic biology is provided. This chapter accomplished and represent the task of knowledge acquisition from experts, user, and literature. The last chapters, describes the system implementation, testing and evaluation. It focuses on the architecture and the knowledge representation for each module in GENETIC BIOLOGY system. This chapter represented part of the design phase (knowledge engineering), in building up an intelligent system. The future task will comprise of all the tasks in development of GENETIC BIOLOGY intelligent system.

Chapter 5
SYSTEM IMPLEMENTATION AND TESTING

The development of intelligent systems, like most conventional programs, inherently implementation, testing and evaluation as part of the development process. System implementation and testing is the following phase after the system design. System implementation is a process that converts the system requirements and design into program codes. Testing is performed to ensure that programs are executed correctly and conforms to the requirements specified. This chapter presents the coding method and testing used during the development of this system. Evaluation is don't in the next chapter.

5.1 DEVELOPMENT ENVIROMENT

Development environment has certain impact on the development of a system. Using suitable hardware and software will not only help to speed up the system development but also determine the success of the project. The hardware and software tools used to develop the entire system are listed in the following section.

5.1.1 Hardware Requirments

The hardware used to develop the system are listed as below:

- Intel Pentium(r) II MMX
- 24MB RAM
- 4.00 GB Hard Disk
- Other standard computer peripherals

5.1.2 Software Requirments

There are a lot of software tools, which are used in designing and writing report. The design process involves the drawing of Semantic Networks , Frames and Rules. During the course of GENETICBIOLOGY development, a number of software tools was used. Table below depicts the software used to develop the system.

Software	Usage	Description
Microsoft Windows 98	System Requirement	Operating System
Visual Prolog 5.2	System Requirement	Coding
Notepad	Database	Build the database to store And manipulate the facts

Table 5.1 Software used to develop the GENETICBIOLOGY system.

5.2 DEVELOPMENT OF THE GENETICBIOLOGY SYSTEM

Since GENETICBIOLOGY is a intelligent system, the language used is Prolog. In this system, the codes written are mostly to convert a string to a list of words by eliminating commas and periods, association of entities, alternative association names, synonyms for entities, words to be ignored, parses and evaluates the query, report the number of solution and report error. (Refer to User Manual in Appendix A).

Development of GENETICBIOLOGY involves endless cycle of testing and modifying the Prolog source code, the language file(GENETICBIOLOGY.LAN) and the database of CELLSYS, DNASYS and GENSYS. Validating the input and then going back to make further changes where necessary.

5.3 DATABASE INTEGRATION

The recent realization that intelligent system technology and database technology are basically complementary has spawned numerous projects in this area [Kerschberg, 1986, 1987]. In GENETICBIOLOGY, the database servers the intelligent system by storing and retrieving knowledge in a suitable form.

An integration of databases with intelligent systems provides a framework for easy maintenance. The database can be used to externalize the parts of the knowledge base that can be modified, deleted and added. This separation makes the knowledge base easier to modify since they are not cluttered by large amounts of declarative facts.

5.4 TESTING THE SYSTEM

Testing is probably the least understood part of the software development project. A bug is any unexpected or undesired aspect displayed by the software being tested. Testing can uncover different classes of errors in a minimum of time and with a minimum amount of effort. In the development of the GENETICBIOLOGY system, few stage of testing are done.

5.4.1 Module Testing

Module testing verifies that the component function properly with the types of input expected from studying the component's design. The first step is to examine the program code by reading through the source code, trying to spot algorithm and syntax error. After that, the code is compared to the requirements of the design in order to make sure that all relevant cases have been considered. Next, the debug is used to view the results and eliminating remaining syntax error if necessary. Finally, feed a predetermined set of data to the component being tested and observe what output actions and data are produced. Each module is tested on its own, isolated form others modules in the system. After each module has been tested, the interaction of these components must be tested again.

5.4.2 Integration Testing

When collections of components have been module-tested, integration testing is the process of verifying that the system components work together as described in the system specifications.

5.4.3 System Testing

The last testing procedure done is system testing. The system test, all implementation aspects of the design. Once the entire system is validated, it must be combined with other system element such as hardware, end-user and databases. System testing verifies that elements are functioning properly and the overall system performance and objectives are achieved which in this case. GENETICBIOLOGY must achieve some kind of reliability, robustness and accuracy.

5.4.4 Analysis of Test Result

In these field trials, GENETICBIOLOGY tutoring were compared and examined by a group of students. Interacting with the system with natural language such as English is very easy to utter query as if asking a tutor. The student agreed the frequent asked questions are able to handle by the system and need further incorporation of other types of questions, so the system can handle all kind of students. This fulfills the project's objective that is to provide some intelligence that is required. (Refer to Sample Outputs in Appendix B).

Chapter 6

SYSTEM EVALUATION AND CONCLUSION

The practical, or technological, motivation is that natural language processing capabilities would revolutionize the way computers are used. Since most of human knowledge is recorded in linguistic form, computers that could understand natural language could access all this information. In addition, natural language interfaces to computers would allow complex systems to be accessible to everyone. Such systems would be considerably more flexible and intelligent that is possible with current computer technology. For technological purposes it does not matter if the model used reflects the way humans process language. It only matters that it works. On the other hand, the present state of knowledge about natural language processing is so preliminary that attempting to build a cognitively correct model is not feasible. Rather, we are still attempting to construct any model that appears to work. [Allen James, 1995].

6.1 RESEARCH OBJECTIVES

The objective of the GENETICBIOLOGY project is to study the combination of various existing techniques in the construction of an intelligent system in an intelligent computer-aided tutoring system. These techniques covers a wide spectrum of knowledge representation methods, primarily rules, semantic networks and frames with special reference to the applications of inheritance, procedure attachment and the integration with database.

6.2 THE THEORETICAL AND PRACTICAL RESULTS OF GENETICBIOLOGY

As mentioned earlier, the GENEBIOLOGY project is conducted using existing theoretical frameworks of intelligent system technology. In this section, the theoretical and practical results accomplished in developing the system are summarized.

6.2.1 Knowledge Engineering

The acquisition of genetic biology knowledge from human experts and other sources and the implementation of this knowledge in the system that is both effective and intelligent, form the groundwork of this research. Applied knowledge engineering techniques are used to develop the framework for the analysis. This includes the formulation of rules within the general approach of the domain area.

6.2.2 Knowledge Representation

This section of the research analyzed and characterized the levels of knowledge used in genetic biology system. Frames, semantic networks and rules provide a convenient structure for representing genetic biology knowledge.

While frames are the preferred means of describing taxonomies of objects, production rules remain the most natural means of describing heuristic problem-solving knowledge. In GENETICBIOLOGY, rules are mainly implemented with the "CLAUSE" which are used to infer to the entity and extract the needed information. They are not build to solve heuristic problem or to generate new facts because unlike in other fields or subjects like chemistry the data keyed in by the user is predefine and precise. For example, in chemistry the knowledge for the production of carbon dioxide is as follows:-

Carbon + Oxygen = Carbon Dioxide or $C + O_2 = CO_2$

So, if a student want to know what is produce by the combination Carbon and Oxygen, he or she may simply type in the word or formulas of the substances in order for the system to reason and give the result. In genetic biology, the data can be represented in various of ways. So, it is best to use natural language to handle the queries.

6.2.3 Dealing With Uncertainties

The solutions to genetic biology queries doesn't require reasoning under uncertainty because it deals with predetermine data. But in GENETICBIOLOGY, error detection function is built to handle inputs which are not found matching to the entity represented.

6.3 SYSTEMS STRENGTH

6.3.1 User Friendly

Since the interaction between the user and the system are in natural language (English) it create a user friendly environment and ease of use.

6.3.2 System Transparency

System transparency refers to the condition where the users do not need to know how is the system structure, its coding and anything related to the system built.

6.3.3 Able to Provide Database and Language Maintenance

Users are able to do housekeeping for database and language maintenance. They can add, delete and update the database and language files.

6.4 LIMITATION

Although, as described above, the GENETICBIOLOGY is successful when viewed as intelligent which able to handle queries of users in natural language. It falls short of the theoretical ideal of duplication an expert's performance. In this section, some of the limitations are discussed.

Like the development of other intelligent tutoring systems, the GENETICBIOLOGY project shares with development of other intelligent systems the problem of constructing knowledge bases. This is often characterized as the knowledge-acquisition bottleneck and it can be viewed from various aspects [Clancey, 1987].

First, developing a knowledge base often involves more just writing down what experts explicitly know. Constructing a knowledge base is primarily a modeling problem, using representational tools like rules and frames that unknown to expert. Expert's models of solving problems can be inherently distinct forms of knowledge. Using a scientific approach such as interviews in constructing the knowledge bases may not be able to capture completely those models used by these experts. Thus, knowledge represented in the system may not be what the experts can articulate.

Second, the system is design mainly to handle procedural knowledge, where queries incorporated in the system are mostly procedural queries, such as:-

- "what is the type of cell".

Although, some declarative type of queries are incorporated, such as:-

- "how does RNA nucleotides produced".

Third, the weakness in the way of queries can presented is limited to only one method. This is due to the time constrains and lack of linguistic knowledge. In this system questions can be asked in form of "ENTITY-ASSOCIATON-ENTITY". For example, the question "Give me the *function of cell* wall". Here the first entity is "*function*" which is associated by "*of*" and followed by the entity "*cell wall*". You can have compound name such as "cell wall" as one entity.

Apart from the limitation discussed above, several other general weaknesses of intelligent system technology remain. These are covered extensively by numerous volumes of literature, principally those by Dreyfus [1972], Weigenbaum [1976] and Searle [1980].

6.5 RECOMMENDATIONS AND FUTURE INVESTIGATIONS

While the research has produced results in a number of areas, there are still numerous research opportunities in this area. In general, additional research is required on at least two fronts:

1. theoretical work on the principles of knowledge acquisition techniques.
2. increase the number of form a queries can be asked.

Knowledge acquisition is a difficult and time-consuming task. It would be extremely valuable if an interface can be used that allows the expert to build the system directly. Such a tool would allow an expert who is not familiar with the intricacies of intelligent system design to build a system. The current available software's have a number of restrictions and are limited to the construction of a straightforward and simple system.

As intelligent tutoring system applications grows in to increasingly complex problem-solving areas, the importance of high-quality user interfaces will increase as well. Although this systems prove to inherent natural language, it do not incorporate all kind of queries as mention in the limitation part. Thus, one significant opportunity for improving the system is to incorporate more form of queries, as listed below:-

- query -> entity (e.g. "cell?" Instead of "type of cell".)
- query -> entity-entity (e.g. "cell structure". Instead of "structure of cell".)

6.6 CONCLUSION

Overall, this project has achieved and fulfilled the objectives and requirement as a intelligent tutoring system as determined during the system analysis. It achieves the intelligent by allowing natural language interaction which able to extract knowledge from database through the rules (CLAUSE).

REFERENCES

Arnold, W.R. & Bowie, J.S. [1986], "Artificial Intelligent: A Personal, Commonsense Journey", Englewood Cliffs: Prentice-Hall.

Auderisk, T. & Auderisk, G. (5th eds.), [1999], " Biology: Life On Earth", Upper Saddle River, NJ: Prentice Hall.

Barr, A. & Feigenbaum, E.A. (eds.) (a), [1981], "The Handbook of Artificial Intelligence", Volume I, Los Altos: William Kaufman.

Bonnet, A. [1985], "Artificial Intelligence: Promise and Performance", Englewood Cliffs: Prentice Hall.

Buchanan, B.G. & Shortliffe, E.H. (eds.), [1984], "Rule-Based Expert System: The MYCIN Experiments of the Stanford Heuristic Programming Project", Reading: Addison-Wesley.

Charniak, E. & Mcdermott, D.[1985], "Introduction to Artificial Intelligence", Reading: Addison-Wesley.

Davis, R. & Lenat, D.B. [1980], "Knowledge-Based Systems in Artificial Intelligence", New York: McGraw-Hill

Duda, R.O. & Reboh, R. [1984], "AI and Decision Making: The PROSPECTOR Experience", in Reitman, W. (ed.).

Ernst, G. & Newell, A. [1969], "GPS: A Case Study in Generality and Problem Solving", New York: Academic Press.

Feigenbaum, E.A. [1982], "Knowledge Engineering in the 1980's", Department of Computer Science, Stanford University, Stanford CA.

Firebaugh, N.W. [1989], "Artificial Intelligence - A Knowledge- Based Approach", Boston: PWS - Kent.

Giarratano, J. & Riley, G. [1989] "Expert System: Principles and Programming", Boston, PWS-Kent.

Goldberg, A. & Robson, D. [1983], "Smalltalk-80: The Language and its Implementation", Reading, MA: Addison-Wesley.

Harmon, P & King, D. [1985], "Expert System: Artificial Intelligence in Business", New York: John Wiley & Sons.

Johnson, P.E. [1983], "What Kind of Expert should a System be?" in The Journal of Medicine and Phlosophy, Vol. 8, pp.77-97.

Keene, S.E. [1989], "Object-Oriented Programming - A Programmer's Guide to CLOS", Reading: Addison - Wesley.
Kleinsmith, L.J. (2nd eds.), [1995], "Principles of Cell and Molecular Biology", New York: HarperCollins.

Luger, G.F. & Stubblefield, W.A. [1989], "Artificial Intelligence and the Design of Expert System", Menlo Park: Benjamin/Cummings.
Martin, J. & Oxman, S. [1988], Building Expert System: A Tutorial", Englewood Cliffs: Prentice-Hall.

Minsky, M. [1975], " A Framework for Representing Knowledge", in Winston [1975], pp. 211-77.

Mockler, R.J. [1989], "Knowledge-Based Systems for Management Decision", Englewoods Cliffs: Prentice-Hall.

Newell, A. & Simon, H.A. [1972], "Human Problem Solving", Englewood Cliffs, NJ: Prentice Hall.

Newell, A., Shaw J.C. & Simon, H.A. [1963], "Empirical Explorations With the Logic Theory Machine: A Case Study in Heuristic", in Feigenbaum, E.A. & Feldman, J. (eds.) [1963].

Pinson, L.J. Wiener, R.S. [1988], "An Introduction to Object-Oriented Programming and Smalltalk", Reading: Addison-Wesley.

Post, E.L. [1943], "Formal Reduction of the General Combinatorial Decision Problem", american Journal of Mathematics, 65, pp. 197-268.

Prepau, D.S. [1990], "Developing and Managing Expert Systems", Reading: Addison - Wesley.

Quillian, M.R. [1968], "Semantic Memory" in Minsky [1968], pp.27-70.

Rauch-hindin, W.B. [1988], "A Guide to Commercial Artificial Intelligence. Fundamentals and Real-World Application", Englewood Cliffs: Prentice-Hall.

Rolston, D.W. [1998], "Principles of Artificial Intelligence and Expert System Development", New York: McGraw-Hill.

Schutzer, D. [1987], "Artificial Intelligence - An Applications-Oriented Approach", New York: Van Nostrand Reinhold.

Shortliffe, E.H., [1976], "Computer-Based Medical Consultations: MYCIN", New York: Elsevier".

Silverman, B.G. (ed.) [1987], Expert System for Business", Reading: Addison-Wesley.

Singapore, [1994],"Computer Analysis of Genetic Macromolecules: Structure, Function and Evolution", Singapore: World Scientific.

Tello, E.R. [1989], "Object-Oriented Programming for Artificial Intelligence", Reading: Addison-Wesley.

Walters, J. & Nielsen, N.R. [1988], "Crafting Knowledge-Based Systems", New York: John Wiley & Sons.

Waldrop, M. M. [1987], "Man-Made Minds: The Promise of Artificial Intelligence", Rexdale: John Wiley & Sons.
Waterman, D.A. [1986], "A Guide to Expert systms", Reading: Addison-Wesley.

Wiener, R.S. & Pinson, L.J. [1988], "An Introduction to Object-Oriented Programming and C++", Reading: Addison-Wesley.

Winston, P.H. & Horn, B.K.P. [1984], "LISP", (2nd edition), Reading: Addison-Wesley.